



A Guide to Selecting Cloud Configuration Management Products

2019

Author: Darren Lasley

Introduction

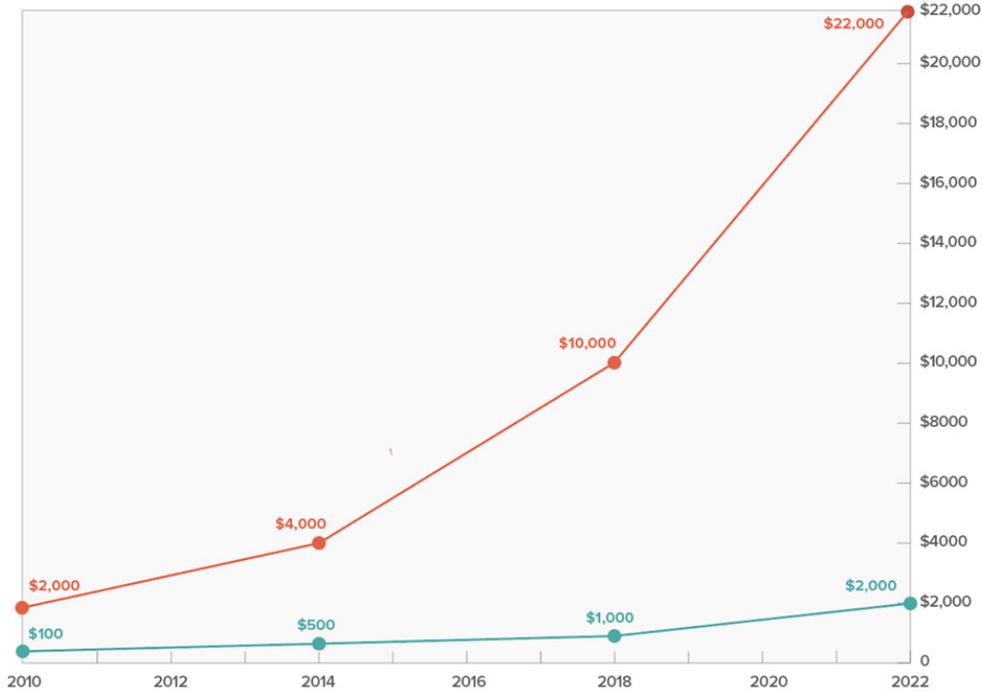
Discovering and addressing the extras needed beyond the [selection of a provider](#) is often overlooked when launching a new product to the cloud. Among the many areas to research are security and compliance tools, centralized monitoring and log aggregation, utilities for backups and snapshots, and anything that automates rote housekeeping. As soon as discovery begins, these are some of the common questions that arise:

- » Where to start looking for options
- » How to analyze and settle on a product
- » Where to find trusted insight and opinions
- » How to validate an option

Research

A common first step is to search the web with a phrase like “Cloud Configuration Management Tools” and begin parsing the results. However, this immediately presents some challenges: Google returns nearly a million results, Bing nearly three million, and so on. The first page of results will undoubtedly have one or two “Top 10” this or “Top 25” that, but often these lists do not provide any clarity towards selection. In fact, the tendency becomes to either hunt through all the results until a particular post or documented use case appears to be a match or, frequently, use an existing vendor relationship to see if any of their offerings may fit the need. While either option could have a positive result, a fast, lightweight process with both rigor and flexibility is more likely to yield true benefit.

Spend Increasing, to Surpass \$22B by 2018



When faced with a critical need and many options, the best place to start is a list of concise, prioritized requirements. Start with the criteria underlying the drivers for a configuration management tool: environment control, audit mechanisms, task automation, etc. If possible, have another stakeholder review them and provide feedback. Repeat with the ideal response for a product meeting the criteria such as “able to provide immediate 30 day view of all configuration changes with time, date, user, workflow, delta, result” etc. These inputs will form the basis of structured analysis of any potential offering.

As a general rule, ignoring the millions of search results will be a good first step when selecting a vendor product. Using a trusted research team, like our own [Level Research](#), will give you a superior view of the market leaders to consider.

Plus, most providers provide insight into their own preferences through [blogs and how-tos](#). In the case of Cloud Configuration Management, keep the focus on vendors’ products which readily extend the capabilities of your provider(s). Also consider the trajectory of your provider(s) relative to their peers—the responses from a vendor may depend on their investment of time and attention to each, often based in turn on the expectations for exposure and revenue growth.

In the Cloud Configuration Management use case, vendors of all sorts and

sizes cover the field. Many are tied to their respective providers e.g. Amazon OpsWorks, Azure Automation DSC, and Google Deployment Manager, while others, such as Platform9 and Google Anthos, are more agnostic. Provider offerings can always be considered, but agnostic options can provide additional flexibility, independence, and an easier path to implementation. Level helps narrow the field by applying our industry expertise to your requirements and identifying the best options for you to evaluate.

Now for the fun part—making the choice.

Analysis

Pull up the requirements and success criteria from the discovery process during Research and incorporate them into an updated comparison chart format. The key here is a balance between covering all bases and remaining nimble—keep the focus on core functionality of configuration management. As a general rule, restrict the list of requirements/criteria to roughly ~30 items to prevent input fatigue and avoid watering down the evaluation.

Next, consider utilizing the flexibility and depth of a scoring matrix to evaluate the capabilities, posture, and trajectory of a vendor and their product. For example:

OUR CURRENT RFIS/RFPS RATING

Functionality	Technology	Composite
30	26	42

VENDOR ASSESSMENT MATRIX

Requirement / Criteria	Category	Vendor Response	Functionality	Technology	Comments
Which IaaS/PaaS providers does your product support? Include planned support if applicable.	Core Product	AWS, DigitalOcean, GCP, OpenStack, RackSpace	3	3	Support for Azure planned for Q4 this year. Currently in discovery with IBM and Oracle.
Which common configuration platforms and languages do you support? Include versions if necessary.	Core Product	Ansible, Chef, Juju, Puppet, Rudder, Salt	4	3	We also have limited support for Synctool and planned support for NOC in next milestone release.

Requirement / Criteria	Category	Vendor Response	Functionality	Technology	Comments
Which companies/ organizations do you consider as strategic technical partners? List in order of their criticality to your product.	Company	Google, Red Hat, Docker, Atlassian, Confluent	3	2	
Describe your product's primary technology pattern and deployment methodology. Include diagrams of systems and flows.	Company	User Interface: WOA and MSA deployed PaaS containers. Config Manager: Mix of EDA on targets, MSA managing all backends.	3	3	
Does your company encourage client-sourced enhancement requests? If yes, describe your inclusion process.	Company	Yes. All requests are reviewed and decided by a management group. Approved minor updates are streamed; major updates are in milestone releases.	4	3	Feature requests are brought in via Zendesk portal, prioritized by frequency and complexity.
Describe your company's ideation and lifecycle frameworks, including time-to-market and time-to-value metrics.	Maturity	We follow a fairly typical customer-focused Agile discipline with user-centric requirements and feedback loops as well as constant iteration around working prototypes.	4	3	See reference section for most recently quarterly TTM & TTV metrics.

Requirement / Criteria	Category	Vendor Response	Functionality	Technology	Comments
How do customers interact with your product from a User Interface perspective? Include any visual representations.	User Interface	All configuration, status, metric, and administration actions can be done via a web-based GUI or python-based CLI.	3	2	See reference section for visual flows.
How do customers interact with your product from an application/ programmatic perspective? Include any API documentation.	User Interface	The primary method is a python-based CLI. Users can also integrate their own scripting calls in java, perl, *nix shell.	3	3	See reference section for current and previous release API specs.
Describe your approach to testing your product against standard security postures (e.g. OWASP) and specific penetration/ exploit point defense scenarios.	Security	All deployed components regularly screened with Coverity. All PaaS components continuously screened with Detectify, Gravityscan, 'others.	3	4	See reference section for most recent security scan report summary.
Describe the authentication mechanisms and patterns you support (oauth, saml-sso, etc.) with examples and specifications.	Security				

Note the opportunity for differential feedback from those involved in selection. Instead of simple Yes/No answers against criteria, objective and subjective analysis converges to achieve clear insight to a vendor's suitability. Level provides additional insight into which questions to ask, how to rank their importance, and how to evaluate the results against the needs of your product or organization.

A couple of notes on this part of the process are relevant:

- » Too many stakeholders can have the same watering-down effect as too many criteria so keep the sample size manageable, no more than 10 if possible.
- » Be transparent through this process so stakeholders remain aware of the competitors, the trending winners and losers, and eventually the actual selection.

Also, remain objective and wary of confirmation bias as feedback is assessed and scoring for the finalist (or finalists) is prepared. When presented with two or three solid options and/or your organization prefers process-driven selection, this may result in the need for committing to an [RFI/RFP](#) which is outside the scope of this document. However, it should be noted that the tools and guidelines above are relevant to that process, especially the collection and evaluation of criteria using both objective and subjective analysis.

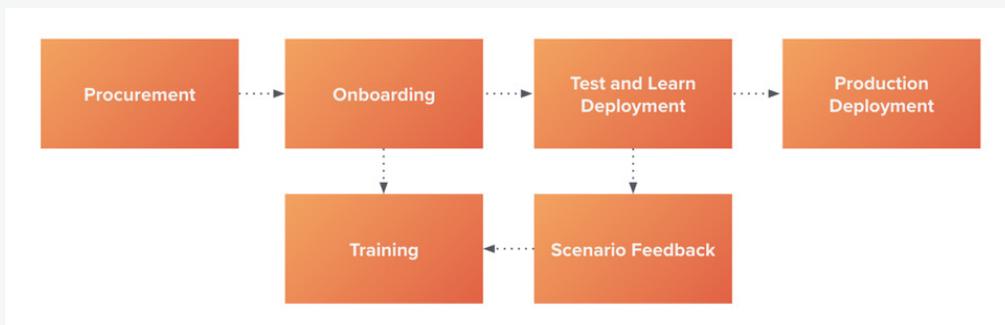
Deployment

To promote a constructive, productive approach when implementing a new tool, consider a disciplined review of the basics: the vendor's typical introduction and first-run documentation, any available GitHub repos, FAQs/KBs involving other customer deployments, and getting a sandbox implementation up and running. The sandbox is essential for giving your teams a chance for early and regular access to a Configuration Management solution supporting:

1. Open and vigorous adoption throughout your organization, critical to consistent use of the configuration management tool, process, and associated standards.
2. Change management, often a heavy process, which is a tightly-coupled companion and needs strong synchronization for both to be effective.
3. Maximum familiarity across teams, essential to identifying the best places and techniques for plugging Configuration Management into your environment.

Per the first item above, do not undervalue the importance of mapping all your intended target systems and processes when identifying standards and crafting processes. For example, the most carefully-written deployment specification cannot overcome a missing object label, a misnamed container, or a keystore without the right client key. Comprehensive support for Infrastructure as Code (IaC) and Configuration as Code (CaC) is an essential characteristic to help address these and other similar scenarios.

Take the opportunity to draft and ratify documents for any standards you need such as server naming conventions, file system locations for configurable items, when to use datastores for configuration, and even a process for introducing change to standards.



Iteration through use cases and observing the interaction with other systems and processes will bring a tool’s eccentricities and behaviors into view. Collect and enact as many use cases as feasible to explore in the sandbox, giving each at least one end-to-end cycle. This is also an optimal time to approach the vendor with any questions or issues, remembering the classic adage “there is no excuse for poor planning, nor does it constitute an emergency for anyone other than the planner.”

Onward

At this stage, all of the pieces are in play: a product, a plan, and a playground. All that remains is the final “P”, Production. Everything up to this point is an exercise in exploring and learning, but the real proof is what happens when users start to interact with the product and systems start to depend on it for second-by-second operation. Some precautionary steps can help make the transition to pProduction easy to digest, among them:

- » Run through the entire deployment plan in a test environment with at least a functional duplication of all production systems and integrations.

- » Use a deployment strategy based on the common [Blue/Green](#) model, allowing pre-release validation as well as rapid rollback in case of issues.
- » If time allows, perform a complete rollback of the test environment deployment and then repeat, looking for any notable differences between iterations.
- » Be retrospective: collect feedback from technical and business users about what went well, what did not, and what could be improved.

With each step, consider recording the inputs and outputs into a documentation repository for reference when building tool and process documentation, an oft-overlooked step which can introduce operational gaps in the future. Having a concise, comprehensive view of how an environment is built and controlled can be just as valuable as a diagram of what exists and where. This is of particular importance in regulated industries where the ability to both demonstrate and provide evidence technology infrastructure controls are often audited.

Conclusion

While this guide may not apply to every vendor or scenario, the process and decision tools can provide the structure and consideration to select a tool which will grow along with products and services deployed in your cloud. Setting aside the time and resources to do the research, perform the analysis, and run some deployment iterations will reap considerable reward when production implementation begins. If you are ready to start the process but unsure about some of the steps—or simply want a partner to assist—let us know how we can help.

About Level

Level helps clients transform their business with strategic consulting and technical execution services. We work with your IT organization, product groups, and innovation teams to design and deliver on your technical priorities.

Level's Cloud Practice combines decades of traditional architecture, development, security, and infrastructure experience with a complete mastery of available and emerging cloud offerings. Our client-centric approach focuses first on understanding your business needs and goals, then selecting the right cloud technology to make you efficient, agile, and scalable. We tailor custom solutions to fit within your business processes, simultaneously reducing TCO and downtime while increasing productivity, security, ROI, and speed to market. For more information, contact hello@level.io

References

<https://451research.com/blog/760-studies-show-managed-services-layer-is-a-major-cloud-service-opportunity>

<https://www.businesswire.com/news/home/20170411005965/en/Global-Cloud-Managed-Services-Market-Grow-17>

<https://www.marketsandmarkets.com/Market-Reports/managed-services-market-1141.html>

<https://blog.usenotion.com/using-objective-and-subjective-data-to-unlock-team-performance-cfae4eedc76c>