



MORPHEUS

5 Steps to Multi-Cloud Mastery and Continuous DevOps Delivery

Last Updated: November 2019

Copyright © 2019 Morpheus Data, LLC. All Rights Reserved
All third-party product and company names are property of their
respective holders and use does not imply any specific endorsement



MORPHEUS

www.morpheusdata.com

ABSTRACT

In today's modern IT organization, the stakes are higher than ever before. If you are reading this then you likely know better than anybody that the pace of digital transformation has put extreme pressure on legacy people, processes, and resources. IT leaders who do not respond effectively to those pressures are likely to be on the outside as a new breed of cloud-savvy and application developer focused individuals come through the ranks.

Organizations in virtually every industry across public and private sector are using software innovation to differentiate and avoid being disrupted. In this new era of digitization, speed and agility are the keys to success and it falls on Infrastructure and Operations (I&O) teams to better enable their stakeholders.

This means requests for resources are expected to be fulfilled at near instant speeds and at higher volume with every passing day. IT touches almost every department within an organization and as such can quickly become a bottleneck as legacy approaches give way to new modalities.

Case in point, it can take IT weeks to deliver new application services to developers which has led many teams to bypass IT completely in favor of public cloud. The resulting Shadow IT and Complexity Crisis has caused a lack of visibility, cost overruns, security holes, brittle infrastructure, and a general breakdown of lifecycle management.

IT leaders need to radically improve speed and efficiency of service delivery to stay relevant and get ahead of new technology transitions such as the shift to Kubernetes, cloud-native PaaS, and serverless application frameworks.

Morpheus is a 100% agnostic cloud management platform (CMP) for multi-cloud orchestration, unified operations and self-service automation.

It's a foundation to bridge the gap between teams, tools, and processes independent of where and how applications are deployed. Unlike vendor specific tools that force you to rip and replace existing technologies, Morpheus unifies tools you already have and lets you manage your clouds your way - just better and with zero lock-in.

This paper will dive deeper into our view of major market trends, discuss 5 critical steps required to transform IT service delivery, and present several considerations to factor in your decision process as you embark on your multi-cloud and DevOps journey.

- Step 1:** Enable Self-Service for All
- Step 2:** Quickly Integrate and Optimize
- Step 3:** Establish Controls and Policy
- Step 4:** Provide Frictionless Delivery
- Step 5:** Manage Day-2 and Beyond

BACKGROUND

By 2021 over 75% of midsize and large organizations will have adopted a multi-cloud and/or hybrid IT strategy.¹

By 2025, 80% of Infrastructure & Operations activities will be in support of digital business products (primarily in the form of applications) rather than the delivery of infrastructure, up from approximately 10% today²

There are two major shifts underway in the technology industry that have forever changed how Infrastructure and Operations (I&O) will deliver on the digital transformation mandate – Cloud Computing and DevOps. While both of these juggernauts are entering their second decade, we are poised as an industry to take greater advantage in 2020 and beyond as they begin to converge and mature.

CLOUD COMPUTING

Cloud has evolved over the last decade from a market disrupting trend to the default mechanism for delivering both traditional and next-gen IT services. At its most simple, cloud computing is defined by elastic scale and self-service consumption. There is no doubt that cloud computing has become an integral part of digital transformation initiatives due to its ability to deliver greater agility and speed of innovation.

Gartner, AWS, and other experts have presented variations of the “5 Rs” construct surrounding the move to cloud. While ‘Retiring’ and ‘Repurchasing’ applications are valid stages they are less critical in this discussion, so we’ll focus on the other three.

- **Rehosting:** Many early cloud adopters chased the illusion that public cloud was cheaper, pursuing bulk migration of applications. Most now agree that was a false belief and we’ve reached equilibrium across private and public deployments.
- **Replatforming:** Involves minor updates to some services and processes as part of moving to the cloud. For example, swapping out a database layer to shift from a legacy RDBMS to a cloud-native PaaS equivalent.
- **Refactoring:** An opportunity to embrace microservices and containers as well as cloud-native features or even serverless functions. Requires more work but can also deliver the greatest set of new innovation to the business.

No matter where they sit on “R-curve”, organizations are pursuing long term relationships with hyperscale cloud providers such as AWS, Azure, and Google. They are looking for continuous access to new innovations but at the same time need to control vendor lock-in by adopting open standards and integration technologies.

¹ Increasing Reliance on Cloud Computing Transforms IT and Business Practices. Gartner Dec 2018 by various

² IT Operations Management 2025: Shift to Succeed. Gartner Oct 2018 by Colin Fletcher and Terrence Cosgrove

This presents a unique challenge for I&O organizations that are already struggling to keep up with the demands on legacy infrastructure and a skills gap which will only get worse as overstretched teams try to keep up with a myriad of cloud interfaces.

FROM DEV | OPS TO DEVOPS

In the last 10 to 15 years, the adoption of Agile Software Development has led to broad awareness and acknowledgement of the IT culture change known as DevOps. Popularized in works such as *The Phoenix Project*³, organizations have been seeking the nirvana of agile methodologies, collaborative feedback loops, and fully automated workflows. It's much easier said than done however, and many have found that people and process often are a harder problem to face than tools and technology.

Still, the continuous improvement of flow by removing constraints and delivering more value is a goal that persists. Adopting DevOps culture, tools, and systems helps the organization to realize more value from IT investments plus deliver software faster and at higher quality. All of that translates to greater market differentiation and business value.

According to the DORA State of DevOps report, there are several stages of maturity and associated practices that are consistently seen as organizations evolve.

- **Normalize and standardize the technology stack:**
This is marked by version control and use of consistent technology stacks and operating environments to reduce variability.
- **Expand DevOps practices and automate delivery:**
This stage is marked by accountability and elimination of manual approvals as well as configuration and provisioning automation to improve flow.
- **Provide shared self-service platforms to scale efforts:**
This highest stage of maturity is marked by self-service for continuous delivery and cross-functional teams to manage feedback loops.

One of the challenges that many have faced in their DevOps journey has been finding tools and technologies that equally meet the needs of both sets of stakeholders. Development teams have gravitated towards agile frameworks, open source tools, programmatic frameworks and infrastructure-as-code, while Operations professionals tend to lean on familiar GUIs, trusted vendor relationships, and service level agreements.

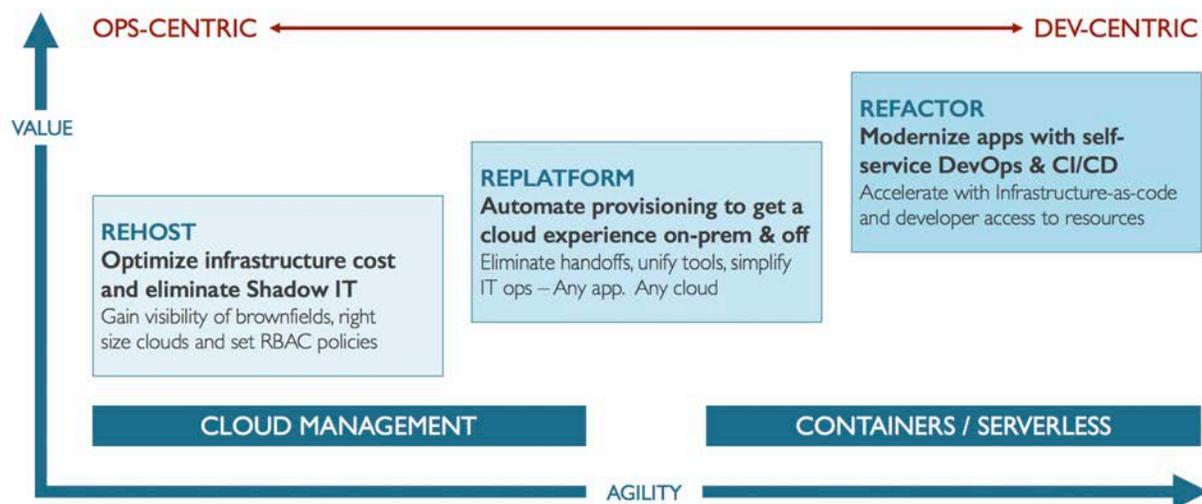
**Bridging on-prem + public cloud as well as Dev + Ops
takes a different kind of tool and a different kind of IT leader.**

³ Gene Kim, Kevin Behr, George Spafford, *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win* (IT Revolution, 2013)

MULTI-CLOUD AND DEVOPS CONVERGENCE IN 2020

Despite Cloud Computing and DevOps having been around for 10+ years, it's clear that most organizations still have a long way to go. It still takes Developers weeks to get what they need from IT delivery and Operations teams are still worried about security and service levels. In addition to the technology trends already discussed, there is an increasing appetite to adopt emerging infrastructure-as-code methodologies and an elevation of awareness surrounding cybersecurity threats.

These factors will only compound the problem unless IT steps up its game, but it's not all bad news. More organizations than ever before have dedicated budget, people, and programs for cloud management and automation. The most forward looking of organizations are actively crossing the streams and rethinking how Cloud Computing and DevOps maturity curves should converge to accelerate business outcomes.



OUR VISION

Morpheus provides a systematic approach to multi-cloud self-service and DevOps automation which bridges the gap between teams, tools, and processes across Mode 1 and Mode 2 applications independent of where and how those applications are deployed.

Secondly, our aim is not to replace existing tools. Instead, we improve the return on those investments by linking technologies together (“the glue for your tools”) and filling in gaps with built-in functionality. We also provide an abstraction so changes in your tools and platforms does not cause disruption to your overall enterprise agility.

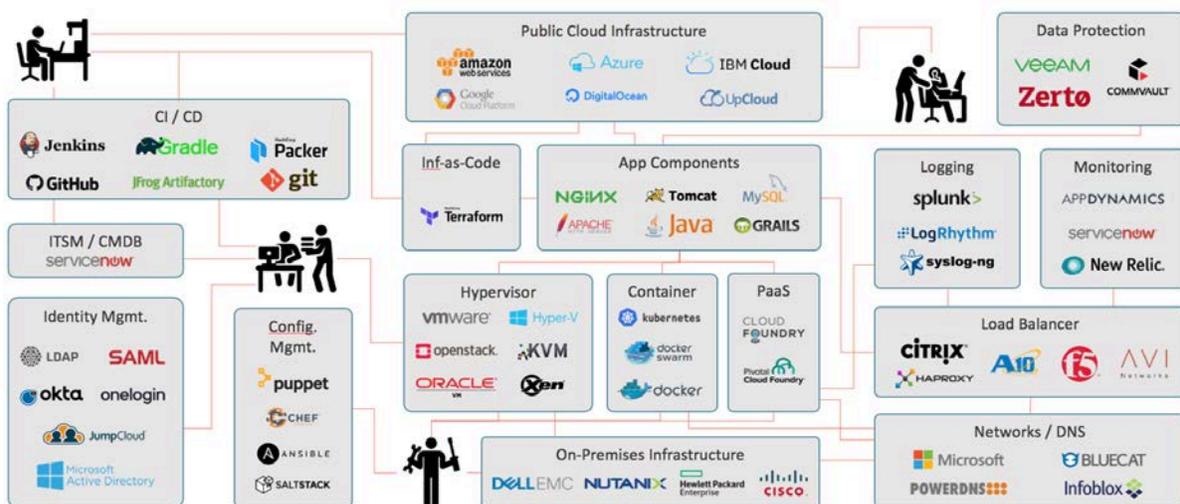
Lastly, we are a unified operations framework that enables Developers, Operations, and Security teams to collaboratively serve the needs of the Business in a frictionless way by delivering on what each needs to be successful without negatively impacting the other.

STEP 1: ENABLE SELF-SERVICE FOR ALL

By 2023, 90% of enterprises will fail to scale DevOps initiatives if shared self-service platform approaches are not adopted.⁴

Let's run through a simple scenario that is repeated hundreds of times a day across virtually all large IT shops. An end-user developer or business user requests a service from IT that involves something as simple as a new app server or Virtual Machine (VM).

In many scenarios this "Service Request" gets added to a long list of already backlogged to-do items for the service delivery team to deal with. An Engineer picks up the ticket and begins the process fulfilling the request. Sounds simple, but depending on the size of the organization, the other departments involved and the degree of interruption due to unplanned work, the complexity of delivery can become exponential in a hurry.



Maybe the Engineer must first open a ticket to the Networking team to request an IP Address. This adds a new delay. The next step might be to request a new Firewall rule be created to enable connectivity to the newly assigned IP. Then a Virtual IP request needs sent to the team responsible for the F5 Load Balancer. More delays and not done yet!

Where does the workload need to run, what apps need to be installed, do backups need to be performed, how about logging and monitoring. Do the right users have permission, has it been joined to the domain, etc. On and on the list of tasks builds, the handoffs pile-up, and time to provision goes from days to weeks.

Couldn't the organization just add more people? The answer is both yes and no. Adding more people does increase concurrency, but to consistently increase flow, concurrency must be increased across all teams. Doubling delivery resources also does not always

⁴ How to Scale DevOps by Building Platform Teams. Gartner April 2019 by Daniel Betts and George Spafford

reduce the time to deliver individual requests. There are inherent delays in handing off tasks between teams and managing queues day to day.

Doesn't the Public Cloud eliminate all those steps? Public cloud does simplify most aspects of provisioning. The ability for developers to quickly self-serve what they need has largely driven public cloud growth in recent years. Unfortunately, 80% of all workloads are still run on-premises, plus concerns over cost overruns and cyber-security have led IT to more tightly restrict public cloud access for development teams.

But we're already doing some configuration management? Individual teams often silo automation by creating small tasks to reduce the work at their station. This view doesn't improve total flow and still requires handoffs. The network team, for example, may use python scripts while the OS team may be using a config management tool. There is no single source of truth and the result is a brittle mess that can't adapt to change.

RECOGNIZING DIFFERENT USER NEEDS

Modern organizations are increasingly complex and as discussed people and process can be one of the biggest inhibitors. That's why Morpheus has been built from the ground up to address part of the people and process conundrum.

Shared understanding is a core tenant of DevOps and it's been manifested in how Morpheus enables unified self-service for CloudOps, DevSecOps, and BizOps teams. The trick is to maintain control within the organizational units for each respective aspect of a service request, while at the same time removing friction wherever possible.

The only way to **improve agility** is to reduce the number of handoffs (or queue transitions) across the workstream. The only way to **reduce risk** is to make sure the success criteria of each Ops unit is represented in the remaining queue transitions. The only way to **increase speed** is to programmatically manage each transition and enable continuous delivery. The only way to **assure adoption** is to enable each team to consume the platform their own way.

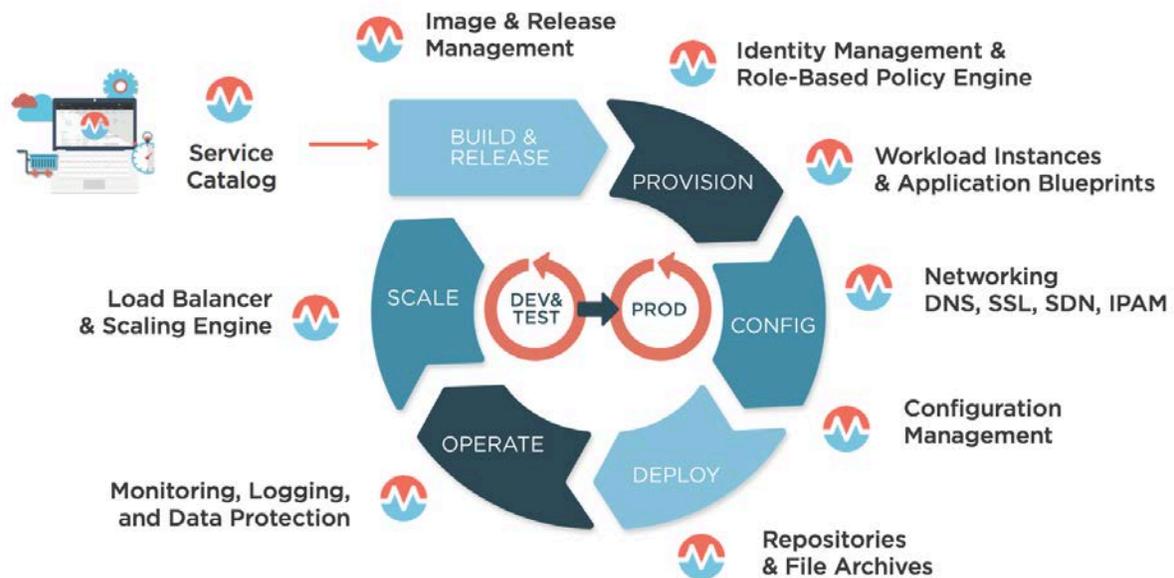
This is exactly what Morpheus enables!

Morpheus reduces delivery time by fully automating each team's request queue while at the same time allowing each team to retain full governance and control of their respective elements.

Morpheus was designed to look at the overall application lifecycle and IT landscape as a whole. The product has been developed with base level abstractions that are broad enough to cover almost any workload concept but also detailed enough to properly represent an object or process within IT. These computing concepts range from Servers, VMs and Containers, all the way to Cloud-Native services.

One of the keys to our ability to keep up with new technologies was the fact that Morpheus' Object model was originally designed using Containers as the base unit of work rather than being limited by a VM-centric architecture. The granular nature of containerized applications and associated processes has enabled the platform to represent an extremely broad range of workload types and will allow us to adapt to Functions-as-a-Service and Serverless without having a major re-write.

In addition to representations across a variety of compute primitives, the platform includes built-in services and abstractions for a wide range of operational tools including ITSM, Certificate Management, IP Address Management (IPAM), DNS, Configuration Management, Backups, Monitoring, Logging, Load Balancers, Workflows and many others. By having written native services for these pieces, we have a unique appreciation for how to integrate best-in-class third party products for each and optimize the linkages between them.



SERVICE CATALOG

Morpheus ships out-of-the-box with 50 or so common application layouts curated for use for all supported cloud types. These *system images* are extremely useful during initial setup and deployment of Morpheus but for production use most enterprises will want to start with a clean slate and load their own IT approved golden images.

User Images are automatically synced from Cloud Integrations and added to the Virtual Images section when a new cloud is attached to the platform. Images can also be uploaded directly into Morpheus via local file or url. Amazon and Azure Marketplace images can also be added to the Virtual Images Section. Lastly, you can use a cloud-native template language such as AWS CloudFormation to provide catalog access to PaaS applications such as DynamoDB and hundreds of other services.

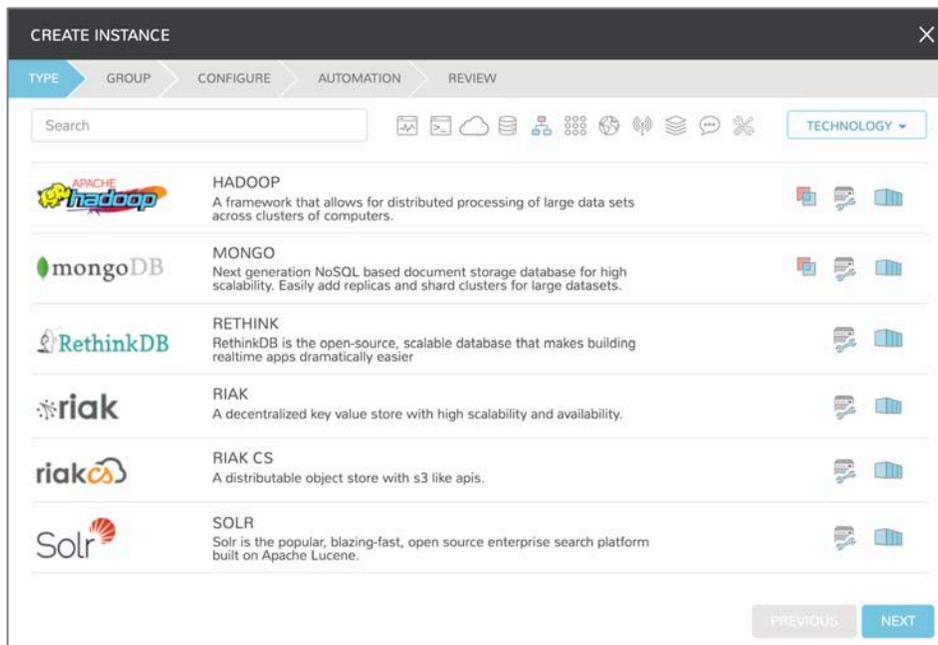
Morpheus also includes a robust Image Builder tool that can be used to create vmdk, qcow2, vhd and raw Images from scratch. The Image Builder creates a blank VM in VMware, attaches an OS iso, executes a boot script on the VM at startup via VNC which calls a preseed script which runs the unattended OS installation and configuration. Morpheus then executes an OVA export of the completed vmdk to a target Storage provider and converts the image to all other specified formats. The new Virtual Image records are automatically added to Morpheus and the Images are then available for use.

Lastly, Morpheus supports hypervisor-to-hypervisor image conversions and extraction of virtual images from certain clouds. For example, Morpheus will create a snapshot of existing VM and upload the snapshot to virtual image directory. Images that have been uploaded to the Virtual Images library can be converted to VHD, QCOW2, RAW and VMDK formats and then re-provisioned into other clouds.

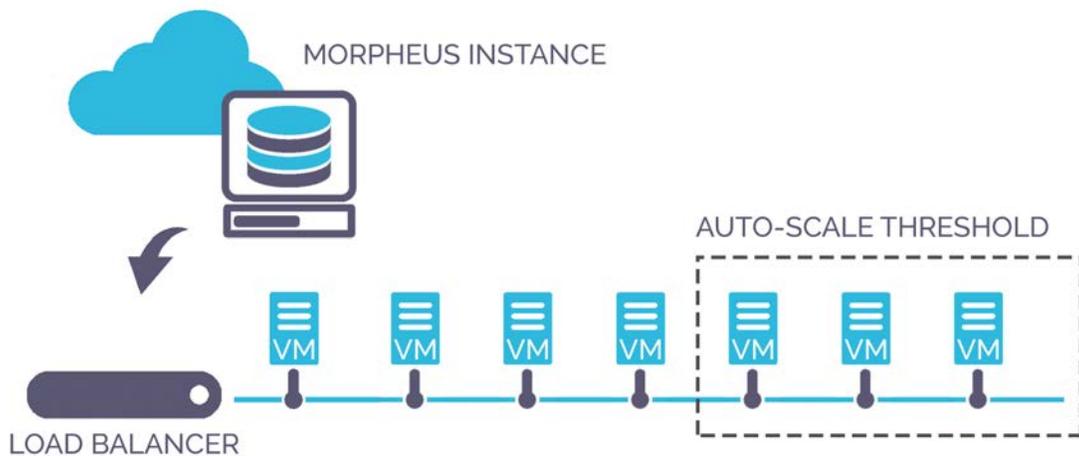
IT'S ALL ABOUT THE APP

Morpheus is a powerful infrastructure agnostic and application-centric automation platform. As a result of this application-first design, there are major improvements compared to legacy infrastructure focused CMPs. Most cloud automation tools from hypervisor and HCI vendors start at the IaaS layer and work up to the application meaning they still need additional PaaS products or added scripting to fill out the gaps.

Morpheus was designed from the application out, making it a more complete platform with greater platform and service delivery capabilities out of the box. Morpheus starts with provisioning individual Instances or 'workload elements'. In some platforms an Instance is representative of a singular object, like a "Virtual Machine" in Amazon. In Morpheus however an Instance is a representation of a Resource or Service. This service may involve several virtual machines or even several docker containers.



For example, in the Morpheus Instance wizard, Mongo is an option and contains several “Instance Configurations”. One of these configurations is a full Mongo cluster consisting of either seven virtual machines or seven docker containers. Rather than representing these directly as seven individual “instances”, Morpheus groups them together into a singular instance of a service that contains multiple containers or VMs. This even allows for instance actions that can be performed to expand capacity either horizontally or vertically. Auto-scaling can be enabled using the built-in Morpheus Scale thresholds and rules based on memory, disk, or CPU utilization and can be applied to scale across clouds for true ‘cloud bursting’. You can also utilize cloud-native technologies like AWS and Azure Scale Sets.

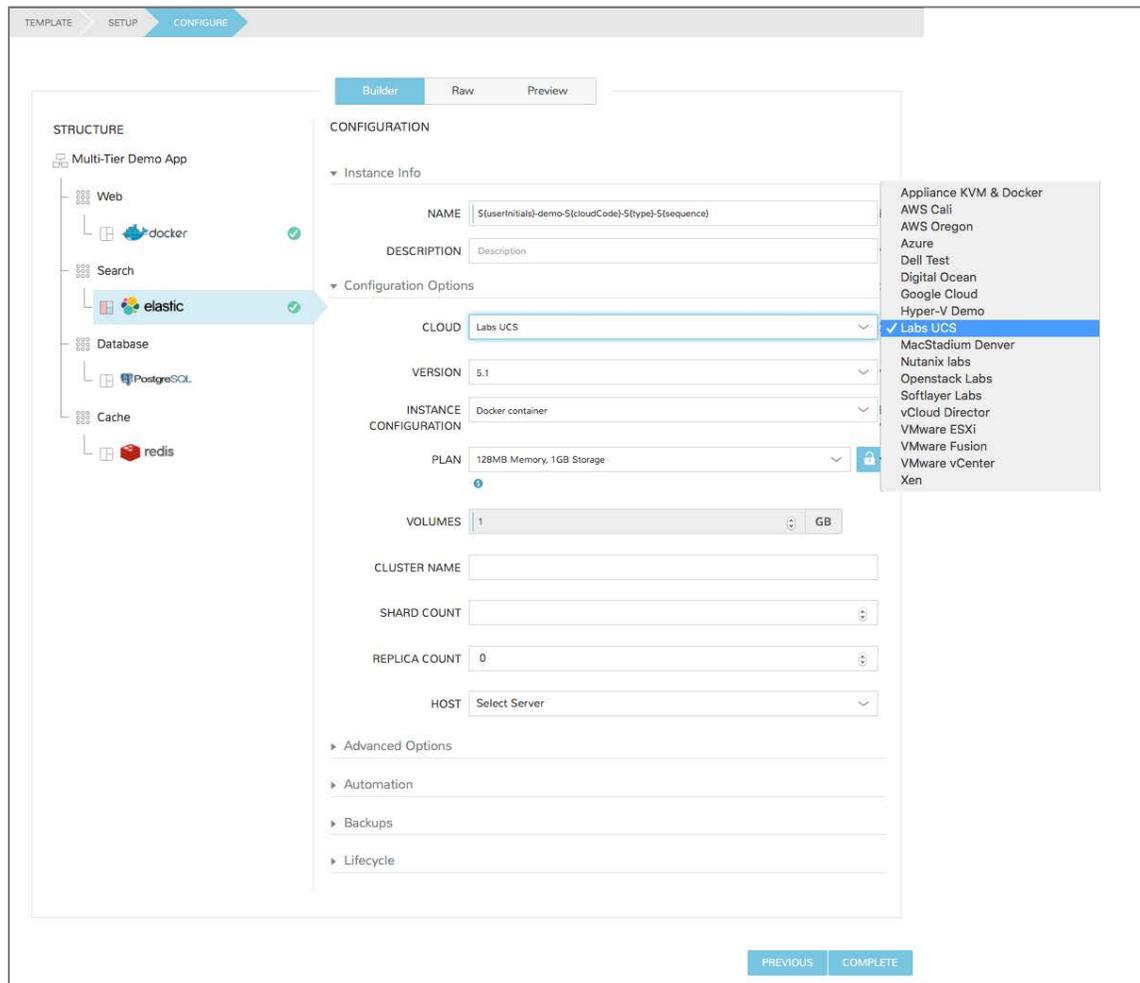


Moving one level down, an instance could have many nodes, a node being a generic representation of a container or a VM. You’ll also see reference to Hosts and Servers. This is a concept tailored to those users maintaining the underlying infrastructure. A Host typically refers to a Docker Host in which a container in an instance is running, or a hypervisor virtual machines can be provisioned onto. A Server is the base representation of a physical or virtual server. It could be a Host, a VM, or even a Bare Metal server.

When a user provisions a VM-based instance, a corresponding server record is created to represent the link to the actual resource. This singular concept is what allows Morpheus to ingest “Brownfield” environments. We do not need to start clean. Morpheus can be integrated into existing environments and manage existing virtual machines. The way Morpheus does this is by periodically syncing existing VMs from the added cloud integrations. A server record will be created and periodically updated (5 minutes typically) with real-time information and changes. This, in essence, provides a rudimentary CMDB and acts as the mechanism to populate third-party CMDBs.

When a server is discovered, the user (given appropriate access) can convert the VM to a managed instance. When this is done, a corresponding Instance is made in the provisioning section of Morpheus and the Morpheus Agent can be installed to provide more refined guest operating system level statistics and logging.

Sitting above the Instance is the concept of an Application abstraction. An App in Morpheus is a collection of Instances linked together via application tiers and associated logic. Tiers allow the user to define segregated sections of connectivity between the various elements / instances within an application. The App can represent boot order amongst tiers and can be reflected in other areas of the platform such as Monitoring for application-level service availability.



We'll talk about it in more detail in 'Step 4: Frictionless Delivery' but this application abstraction is quite flexible and powerful. Application stacks can be represented in YAML/JSON and can be fully provisioned from the API as well as build tools like Jenkins to achieve continuous delivery. Morpheus also supports third party application templating syntax including Terraform, CloudFormation, ARM, Kubernetes Spec, and HELM.

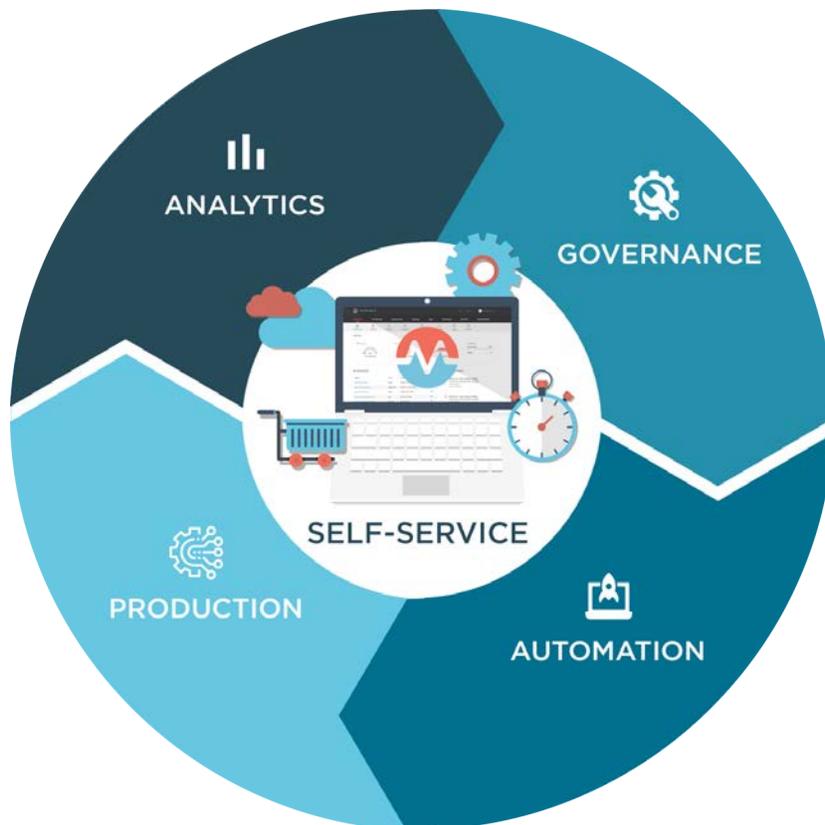
This means whatever tools and process your developers are already using we can add operational security, policy, and consolidated cloud management without slowing things down. It's agility without anarchy!

WHERE DO WE GO FROM HERE?

Self-Service is the core Morpheus design center which is why it's the first step in the process. However, it's not possible to execute a fully mature Multi-Cloud and DevOps strategy without addressing other elements.

Morpheus does more than simply bolt-on these features as separate products or afterthought modules. With an all-inclusive feature set and consolidated engineering team, we've maintained application-centric view of enabling CloudOps, DevSecOps, and BizOps teams in a uniquely systematic process.

Read on to see how Morpheus applies Analytics, Governance, Automation, and Production automation to build on this Self-Service foundation plus provide a futureproof path evolve from yesterday to today to tomorrow



STEP 2: QUICKLY INTEGRATE AND OPTIMIZE

Most organizations today are running a mix of at least 3 platforms/clouds with new options constantly being added, particularly in industries that are rife with mergers and acquisitions. Speed of integration is also important given the constant change in the surrounding tool and technology ecosystem. For this reason, a modern cloud automation platform must be able to do well in messy and mixed environments.

If it takes you 4 to 6 months to fully roll-out script intensive cloud automation, then you are likely to never realize the benefits of automation. That's because by the time you are done, you'll need to start again. These legacy approaches are brittle and unable to cope with the variety and velocity of today's rapidly changing cloud landscape.

RAISING THE BAR ON INTEGRATION

We live in a software-definable world today full of rest-APIs and web-hooks; notice we say software-definable rather than software-defined. Most of the organizations we work with have dozens of infrastructure components and tools that are loosely connected by 'human middleware'. Only by linking together existing tools and technologies can you enable an on-prem cloud experience and achieve a software-defined datacenter. Many legacy VM cloud management tools advocate web-hook architectures which enable them to call third party APIs, if you, the customer, write a bunch of application programming logic. This is a very liberal use of the word '*integration*' and not one that Morpheus agrees with.

When Morpheus says it integrates with over 80 different third-party products out-of-the-box (OOTB), it is a first-class integration.

Want to add an existing VMware cluster and turn it into a private cloud? Want to use Infoblox as your IPAM tool? Simply click the 'Add Integration' button on the component, enter the appropriate credentials, select some default information and that's it.

This is the reason we are often able to take customers farther in a few hours than a team of consultants from VMware or Cisco can do in that many months.

EDIT CLOUD [X]

NAME: VMware Demo Cluster

CODE: vmwareDemo

LOCATION: []

VISIBILITY: Public [v]

TENANT: Morpheus [v]

ENABLED

AUTOMATICALLY POWER ON VMs

Details

API URL: https://10.30.21.180/sdk

USERNAME: administrator@vmware.bertramlabs.com

PASSWORD: []

VERSION: 6.7+ [v]

VDC: labs-denver-demo [v]

CLUSTER: All [v]

RESOURCE POOL: All [v]

RPC MODE: Vmware Tools [v]

HIDE HOST SELECTION FROM USERS

INVENTORY EXISTING INSTANCES

ENABLE HYPERVISOR CONSOLE

KEYBOARD LAYOUT: en-us [v]

▶ Advanced Options

▶ Provisioning Command

SAVE CHANGES

It's also why we use the word 'cloud' differently than some. If we assume that cloud equals an elastic and on-demand resource, then we want to help you enable a cloud from your existing on-prem stacks as well as your public clouds - VMware, Nutanix, Hyper-V, OpenStack, Cisco UCS, HPE OneView, AWS, Azure, Google, Alibaba, and more.

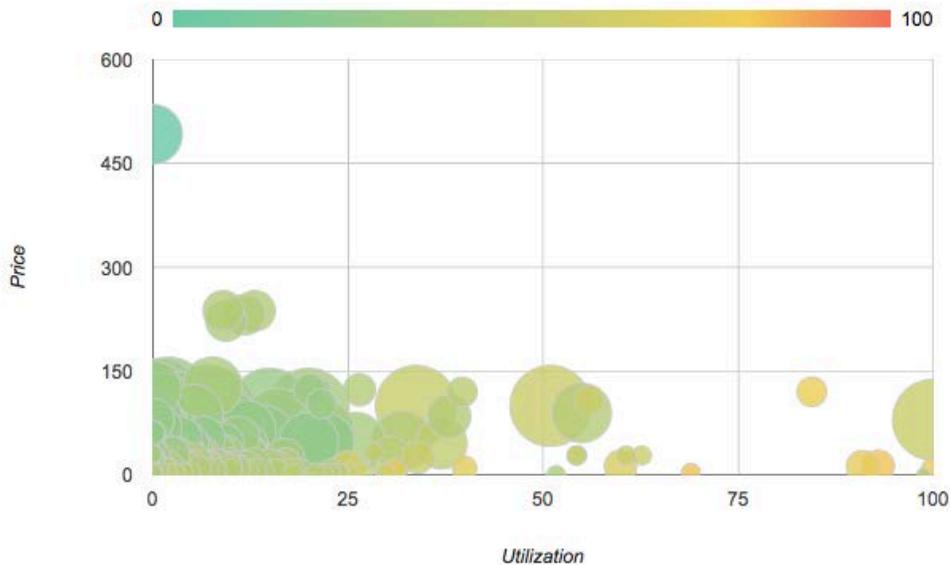
ENABLING EXTENSIBILITY

Not only do we have the largest set of OOTB integrations of any CMP, we also have the most robust set of task types so you can use our workflow engine to augment our integration matrix. Task types include Ansible, Bash, Chef, Groovy, HTTP, Javascript, jRuby, PowerShell Puppet, Python, SSH, vRealize Orchestrator, and WinRM.

IMPROVE VISIBILITY AND OPTIMIZE CLOUD COSTS

For both Brownfield and Provisioned instances, the Morpheus Analytics engine analyzes resource utilization and costs across clouds. This functionality helps users make decisions on where instances and workloads should be provisioned plus assures you are able to keep costs under control.

This example analytics report is a good example of the operational intelligence available through Morpheus. For example, items in the top-left of this chart are very expensive yet very underutilized. Probably something to check-out. On the other hand, items on the bottom right don't cost that much but could be starved for resources and causing end-user frustration or performance challenges.



Not only can Morpheus provide robust reporting of all instances and usage across all clouds, it can also execute rightsizing actions via Morpheus Guidance to get costs back in line. Industry averages show that across public clouds and more efficient use of on-prem infrastructure, you can lower cloud costs by 30% or more.

STEP 3: ESTABLISH CONTROLS AND POLICY

One of the main reasons cloud cost control has been a problem for organizations is a lack of effective role-based access control (RBAC). The ability to provide self-service and automation while still assuring proper guardrails and policy is the core use case for most Morpheus customers. If self-service provisioning is the heart of Morpheus, then the governance engine is the nervous system.

The RBAC and Multi-Tenancy model within Morpheus is very flexible and is designed to work well within a wide variety of implementations. This discussion will provide an introduction to critical RBAC concepts which can then be applied in different ways to reach the desired governance model.

- **Tenants:** A Tenant in Morpheus is an isolated environment with unique users and workloads. A Master Tenant can share resources with Sub Tenants but Sub Tenants cannot share resources or see resources with other Tenants. Morpheus can even add multi-tenant sharing to typically single-tenant platforms like vSphere, Nutanix, etc.
- **Roles:** User Roles determine Feature, Group and Instance Type access for all Users across the GUI, API, and CLI. Entire sections within the UI can be hidden based on the specified access levels. Morpheus supports authentication and role mapping to identity sources such as LDAP, Active Directory, SAML, OKTA, etc.
- **Groups:** Define what resources a User has access to based on their Role. Clouds are added to groups, and a User can only access the Clouds that are in the Groups their Role(s) gives them access to. Resources such as Networks, Datastores, Resources Pools, and Folders have additional Group access settings.

The screenshot displays the Morpheus DevTeam interface. At the top, it shows 'DevTeam' with 'EDIT' and 'DELETE' buttons. Below this, it indicates 'Clouds: 4', 'Servers: 1', and 'Created: 12/06/2017 03:57 PM'. The interface has tabs for 'HOSTS', 'VIRTUAL MACHINES', 'BARE METAL', 'CLOUDS', and 'POLICIES', with 'CLOUDS' currently selected. Under 'ACTIVE CLOUDS', there is a table with columns: TYPE, NAME, LOCATION, HOSTS, VMS, BARE METAL, and a trash icon. The table lists four active clouds: AWS (US West (N. California), 1 host, 0 vms, 0 bare metal), Nutanix Cluster 2 (denverDc aos 5.9, 0 hosts, 0 vms, 0 bare metal), UCS Docker & KVM (Denver, 0 hosts, 0 vms, 0 bare metal), and VMware Demo Cluster (0 hosts, 0 vms, 0 bare metal). Below the active clouds, there are sections for 'PUBLIC CLOUD' and 'PRIVATE CLOUD'. The 'PUBLIC CLOUD' section shows 'Alibaba Cloud' and 'amazon web services' with '+ ADD' buttons. The 'PRIVATE CLOUD' section shows 'Microsoft Azure Stack' and 'CLOUDFOUNDRY' with '+ ADD' buttons.

TYPE	NAME	LOCATION	HOSTS	VMS	BARE METAL	
amazon web services	AWS	US West (N. California)	1	0	0	🗑️
NUTANIX	Nutanix Cluster 2	denverDc aos 5.9	0	0	0	🗑️
UCS	UCS Docker & KVM	Denver	0	0	0	🗑️
vmware	VMware Demo Cluster		0	0	0	🗑️

Once the RBAC structure is in place and Clouds have been added into Groups, the Morpheus policy engine can take over. Policies are used to provide enhanced governance, ease of use, cost-savings, and auditing features to Morpheus.

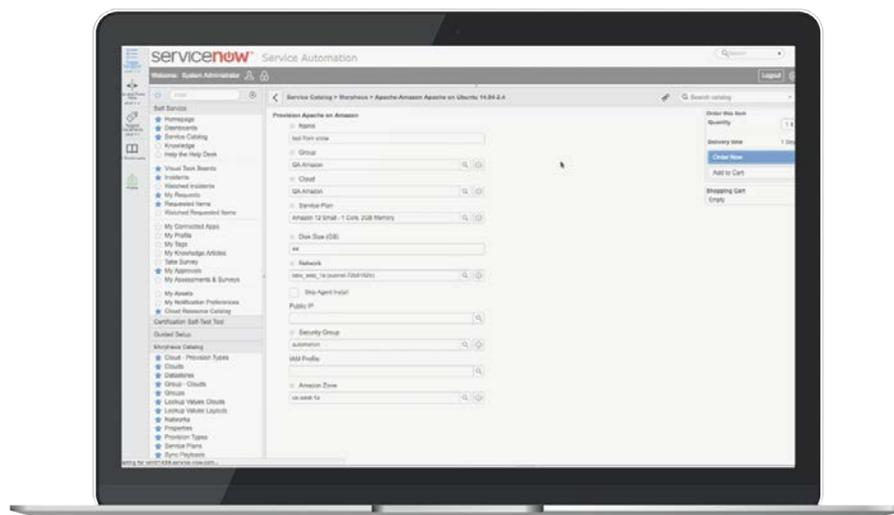
Policies can be scoped to Users, Roles, Groups, Clouds, Tenants and Global scoping to give Admins full control of their environments both on-prem and in the public cloud. It's a control freak's dream come true! Policy Types include (but not limited to):

- **Backup Creation:** Disable / enable creation of a backup when provisioning
- **Budget:** Sets a maximum total combined price for all instances
- **Expiration:** Timeframe in days after which the Instance will be deleted
- **Storage Quota:** Limit storage usage (in GB) for file, object, or volumes
- **Naming:** Fixed or editable name using $\${variable}$ patterns and/or text.
- **Max Containers / Cores / Hosts / Memory / Storage / VM:** per Group or Cloud
- **Power Scheduling:** Adds a Power Schedule to auto shutdown or turn on Instances
- **Provision Approval:** Require internal or third-party approval such as ServiceNow
- **Shutdown:** Days upon provision after which the Instance will be stopped.

INTEGRATION WITH IT SERVICE MANAGEMENT (ITSM)

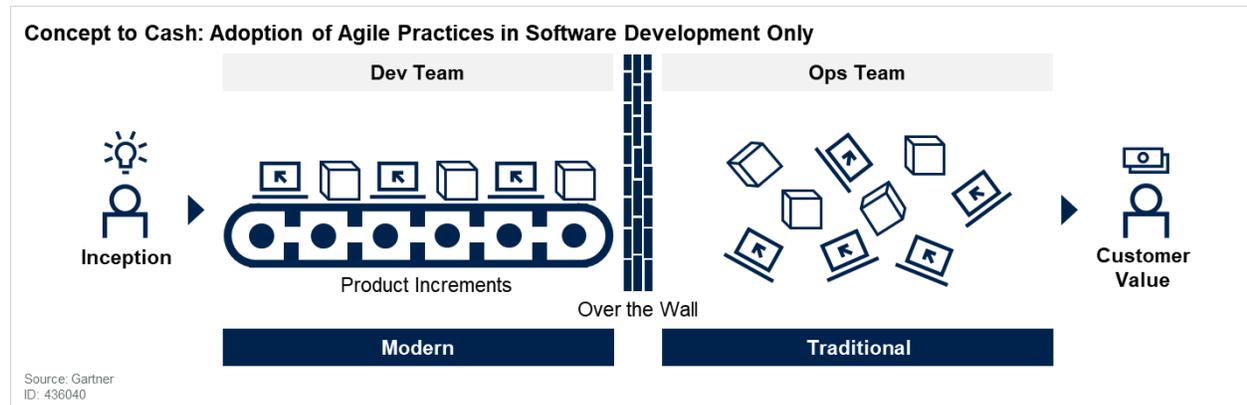
ITSM refers to the entirety of activities – directed by policies, organized and structured in processes and supporting procedures – that are performed by an organization to design, plan, deliver, operate and control IT services.

Morpheus has extremely deep integration into ITSM tools out-of-the-box, including ServiceNow (SNOW), BMC Remedy, and Cherwell. For example, SNOW users can enable provisioning from the SNOW UI via a fully certified SNOW plug-in, plus leverage built in capabilities for CMDB updates, incident management, and workflow approval.



STEP 4: PROVIDE FRICTIONLESS DELIVERY

In a recent Gartner survey of enterprise organizations, they found that accelerating product delivery was the No. 1 objective for adopting agile methodologies. However, accelerating processes on the software development side of the equation, without similarly driving speed in the release of those features into production is counterproductive. This is one of the major forces pushing towards DevOps culture change, but it can be one of the more challenging to deliver if you are solely focused on command and control. To put it another way, Governance is important, but it cannot be at the expense of velocity and innovation.



Source: *Maximize the Success of Enterprise Agile: Utilize DevOps as the Accelerator*. Gartner, Inc. September 2019

Unlike traditional Ops-centric multi-cloud management platforms, Morpheus was designed with DevOps delivery in mind. It was also designed to fit into existing development tool chains and methodologies with little to no disruption. If you are just starting out on a DevOps journey or are looking for a next-generation platform to help scale established processes Morpheus can deliver.

A BLUEPRINT FOR SUCCESS

Morpheus App Blueprints allow pre-configured full multi-tier application deployments for multiple environments and can be triggered as part of CI/CD workflows. They support a vast array of providers and configurations with programmatic markup or Infrastructure as Code capabilities. These flexible multi-cloud configurations can be manually added or scoped to a git repo to help maintain version control and enable teams to work more effectively.

Morpheus supports a number of blueprint template types so that teams can get up and running quickly without having to change their underlying technology stack. Supported blueprint types include Morpheus, Terraform, ARM (Azure), CloudFormation (AWS), Kubernetes Spec, and HELM.

A unique capability of the YAML/JSON based Morpheus blueprint structure is the ability to have multiple configurations per instance within the blueprint. For example, maybe the "development" environment doesn't need as many horizontally scaled nodes as the

DEPLOYMENTS

The deployments section of Morpheus provides very useful PaaS like capabilities when it comes to deploying application artifacts into the newly provisioned environment. These can be uploaded directly from the UI, pulled from a build server, pulled from a public or private Git repository, or even via the API and the various plugins created, such as Jenkins, and Gradle to support continuous build / integration workflows.



A deployment can be considered a set of versions that relate to a particular project or application being deployed. This allows one to keep track of a history of versions and easily reuse these deployment versions across instances that may exist in different environments. An example might be to deploy a version from a deployment to a staging instance and (once approved) also deployed into production.

Getting started with deployments is easy. They can vary slightly for the application stack being deployed, but the simplest phase of a deployment is adding a version and adding the appropriate files to the deployment archive that are needed for the application to run. This could be a single file like a WAR file for Tomcat, or it could be hundreds of files for stacks like Ruby on Rails. For performing git-based deployments, Morpheus supports both public and private repositories. Fetch based deployments are also very straightforward. Simply enter a URL to a file representing the deployment. This can be a single file (in which case it will just be added to the deployment archive singularly) or it can be a zip file (which will automatically be expanded into the archive).

Additionally, when utilizing Morpheus deployments, scripts can be executed as part of a pre-deployment, post-deployment workflow for any additional configuration necessary to make sure the artefact is made active on the target.

STEP 5: MANAGE DAY-2 AND BEYOND

As an industry and as IT professionals, we spend a lot of time talking about automation and creation of new services. However, a vast majority of time, money, and pain is related to the downstream care and feeding of enterprise application libraries.

Morpheus not only provides acceleration for provisioning, it enables complete lifecycle management of application instances from creation to teardown, as well as tools to help users gain self-service access to operational items such as logging, monitoring, and backup.

TASKS, WORKFLOWS AND JOBS

As mentioned earlier, Tasks Types include bash, powershell, http/api, ansible, chef, puppet, SaltStack, groovy, python, jRuby, javascript, vRO and library scripts and templates, which can be configured for resource, remote or local execution targets.

Workflows come in two flavors and are used to chain Tasks, pass variables, collect user inputs, and mix task technologies. *Provisioning Workflows* are phased based executions attached to individual application instances when they are provisioned. *Operational Workflows* are 'one-off' automation tasks that can be run on a schedule or ad hoc basis for a myriad of day-2 tasks including housekeeping, patching, etc.

The screenshot shows the 'EDIT WORKFLOW' dialog box for a provisioning workflow. The 'NAME' field is 'CreateDemoAppDB', 'DESCRIPTION' is empty, and 'PLATFORM' is set to 'All'. Under the 'Tasks' section, there are two main categories: 'Pre Provision' and 'Post Provision'. The 'Post Provision' section is expanded, showing a search bar and a list of two tasks: '1 DemoAppDDL' and '2 RunDemoAppDDL'. Other sections include 'Start Service', 'Stop Service', 'Pre Deploy', 'Deploy', 'Reconfigure', 'Teardown', and 'Permissions' with a 'VISIBILITY' dropdown set to 'Private'. A 'SAVE CHANGES' button is at the bottom right.

The screenshot shows the 'EDIT WORKFLOW' dialog box for an operational workflow. The 'NAME' field is 'Go Do Some Work!', 'DESCRIPTION' is 'Demo Operational Workflow', and 'PLATFORM' is set to 'Linux'. Under the 'Tasks' section, there is a search bar and a list of three tasks: '1 Add User task', '2 python bar', and '3 pssh-command GETCOMMAND'. Below the tasks are sections for 'Option Types' (with a search bar and 'Project ID test' task), 'Permissions' (with 'VISIBILITY' set to 'Private'), and 'Advanced Options' (with an 'ALLOW CUSTOM CONFIG' checkbox). A 'SAVE CHANGES' button is at the bottom right.

Jobs are for scheduled execution of Tasks and Workflows. Jobs can be set to execute on a schedule and/or execute manually. Jobs are linked to existing Tasks or Workflows and also allow for custom configuration options. Jobs can be associated with Instances, Servers, or have no association, such as a job for a ssh task.

MONITORING

Anything provisioned within Morpheus automatically gets a 'check' created in the monitoring service. These checks are organized hierarchically in "Groups" and "Apps". This makes it easy to gain a perspective as to what a customer or full stack facing impact is in the event of a particular instance failure. This also takes into account redundancy layers when it comes to calculating the applications overall uptime percentage.

One interesting thing about these checks is they are service-type aware. There are several different built in check types that are selected based on the service or instance type that is being provisioned. These range from database type checks to web checks and message queue checks. They are highly configurable and also feature fallback check types for more generic use cases.

NEW CHECK

CHECK INFO SSTUNNELING

NAME

HOST Hostname

PORT 3306

TYPE MySQL Check

DB NAME My Database

INTERVAL 5 Minutes

USER username

PASSWORD

MAX SEVERITY Critical

Max Severity

Affects Availability

MySQL

Check Criteria

QUERY

OPERATOR Less Than

CHECK RESULTS

SAVE CHANGES

There are also several integrations built into the monitoring subsystem of Morpheus including App Dynamics, New Relic, and Service Now. These integrations enable us to automatically install agents for these platforms as well as raise incidents and alerts.

There are several ways to configure alerts and notifications within Morpheus. Users can be notified via Email or SMS as well as several other direct integrations including PagerDuty, Alert Ops, Victor Ops, and even Slack Channel notification.

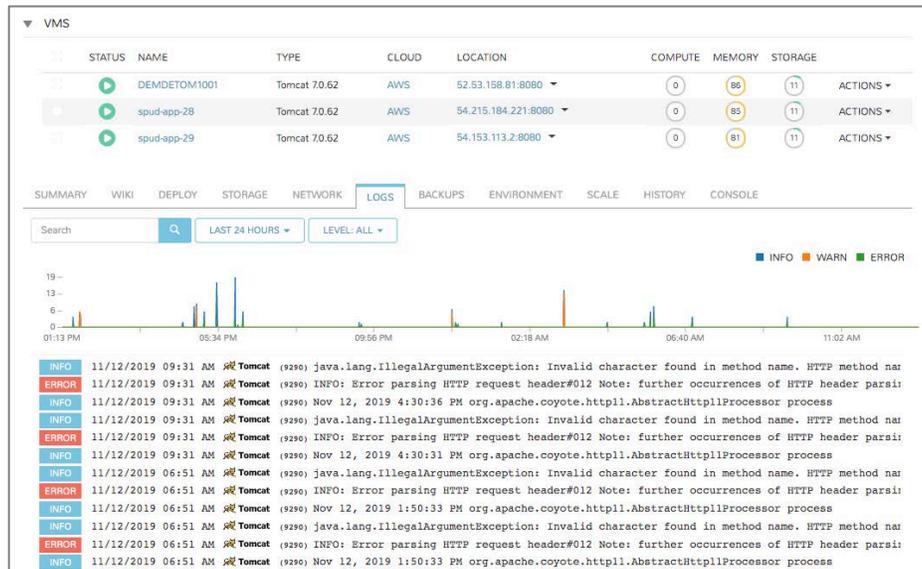
LOGGING

The logging architecture backing Morpheus uses the latest and greatest technologies and standards to be able to service large amounts of log traffic, as well as facilitate easy viewing. Utilizing Elasticsearch behind the scenes and buffered log transmission protocols, Morpheus provides a highly efficient and highly scalable solution for capturing log data from anything provisioned via the system. By utilizing common formats (syslog) it is also very easy to forward logs to external third-party log services.

Morpheus automatically sets up and configures logging for all of the standard catalog items provisioned through Morpheus. This includes both containers as well as virtual machines.

Simple view instance specific logs in instance detail via the “Logs” tab. Logs are also available for items provisioned and maintained by the Morpheus Cluster engine. For example, you can capture logs for Master and Worker nodes associated with the Morpheus Kubernetes Service.

While the built-in logging solution provided by Morpheus is sufficient for most, there are some scenarios in which a more advanced logging system may be desired or already in place. To facilitate this, Morpheus makes it easy to add custom syslog rules as well as built in direct integrations with Splunk and LogRhythm.



BACKUPS

Morpheus built-in Backup solution provides VM, Container, Host, Database, File, Directory, Volume and Storage Provider Backup, Snapshot and Replication capabilities. Backups can be automatically configured during provisioning or manually created at any time. Backup Jobs with custom Execution Schedules and retention counts can be created and used across all environments in conjunction with configured Storage Providers. Backups can be restored over current Instances or as new Instances.

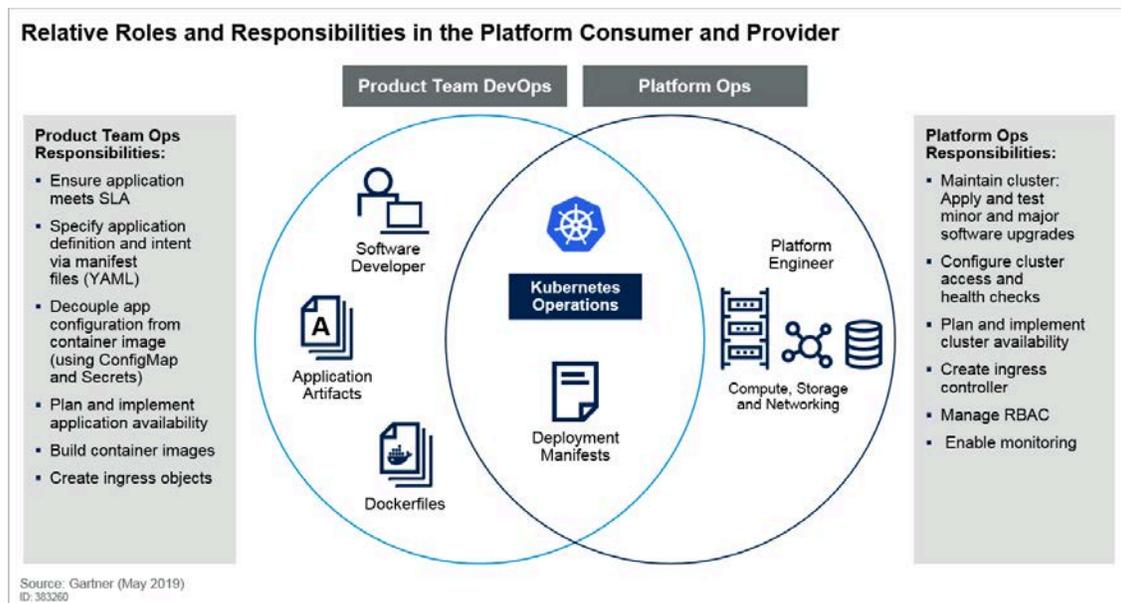
Morpheus also integrates with external services to automate availability with other providers including Commvault, Veeam, Rubrik, Zerto, and Avamar.

FROM VM TO KUBERNETES TO WHATEVER'S NEXT

Morpheus has been enabling cloud-native operations and DevOps processes since the beginning. We've been automating provisioning to docker containers for years including our own Morpheus Orchestrator as well as Docker Enterprise and Kubernetes. When we got our start in 2015 most enterprises had little understanding of containers but today it's becoming a core part of application modernization strategies.

By 2023, Kubernetes will be deployed in more than 80% of all on-premises private cloud and cloud inspired environments, up from less than 10% today, requiring I&O to acquire operational competencies.⁵

Gartner research indicates that 20% of global companies are now running containers in production and while developers have fully embraced the platform, I&O leaders have found Kubernetes to be difficult to install, operate, and maintain. The same IT teams trying to manage on-prem VMware and Public Cloud are now being tasked with Kubernetes projects which is in turn exposing major skill and resource gaps.



BUILD, MANAGE, AND UTILIZE KUBERNETES CLUSTERS ON ANY CLOUD

While some IT teams may have the skills to 'build your own Kubernetes' (BYOK), it can be quite time consuming to manage plug-ins and integrate service layers. Morpheus ships out-of-the-box with a Morpheus Cluster service which can be used to quickly stand up our CNCF-certified Morpheus Kubernetes Service (MKS) on any one of the dozen or more clouds we support. This approach gives you a consistent and fully-managed Kubernetes service independent from the variations found on various cloud platforms.

⁵ How I&O Can Properly Deliver Kubernetes Support. May 2019. Dennis Smith

CLUSTERS

Search

+ ADD CLUSTER ▾

STATUS	TYPE	NAME	LAYOUT	WORKERS	COMPUTE	MEMORY	STORA
✓	kubernetes	davids-k8s	Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	3	3	17	7
✓	docker	demoDockerVMware	VMWare Docker Ubuntu 16.04	1	0	6	11
!	docker	docker-demo-aws	Amazon Docker Host	3	0	37	0
✓	docker	docker-demo-nutanix	Nutanix Docker Ubuntu 16.04	1	0	18	27
✓	kubernetes	k8s-demo-aws	Kubernetes 1.14 Cluster on Ubuntu 16.04, Weave, OpenEBS	3	4	31	30

Kubernetes Cluster
 Docker Cluster
 KVM Cluster
 Triforce Cluster
 EKS Cluster

Like all facets of Morpheus, we want to provide a native service but also integrate with third-party tools so you don't have to be locked into any one path. Our Cluster Service can also be used to quickly provision and manage other services such as Docker Clusters, Amazon Elastic Kubernetes Service (EKS) and Azure Kubernetes Service (AKS). For those that may still want to customize various services within their cluster deployment, you can use our Image Library and modify our Kubernetes layout to fit your needs while still assuring an automated and repeatable deployment.

With Morpheus you can:

- Share K8 clusters across tenants and teams
- Add and manage Namespaces including service plans
- Monitor and control K8 Master and Worker nodes
- View details on specific services including labels and specs
- Monitor status and specs on specific Containers
- Get full details on Jobs, Volumes, Data Stores, and more

With version 4.2 of Morpheus, we'll extend this service to include full inventory and utilization of brownfield Kubernetes clusters meaning if you are already using a popular technology like Red Hat OpenShift you can incorporate those clusters and enable a more open and affordable approach to Kubernetes.

Building and managing the Kubernetes cluster is only a small part of the equation. This is where the rest of the Morpheus stack really starts to shine. We can apply all of the multi-cloud and DevOps services already discussed in this paper to your greenfield and brownfield Kubernetes clusters. Self-service provisioning, cost optimization, secure governance, and day-2 automation.

The result is a unified but still 100% agnostic approach to both VM and Container application management and automation.

CONCLUSION

Wherever your organization is on their multi-cloud and DevOps automation journey and whatever your role, Morpheus can provide a platform to help you go further, faster.

By linking together all of the tools and technologies in a way that acknowledges the people and process challenges of modernization, you can improve control and agility at the same time. We hope this introductory guide has served as a tasting menu for the art of the possible when it comes to the Morpheus platform.

There is no single product or magic bullet that can make your Digital Transformation initiative an overnight success, but we believe our agnostic automation and orchestration platform can be a catalyst to help you get there faster and provide the freedom to more quickly adapt to an ever-changing landscape.

RESOURCES

- For additional detail on any of the features or capabilities covered in this paper you can check out the full Morpheus documentation site at docs.morpheusdata.com.
- To get a third-party view of this market check out the Gartner 2019 Magic Quadrant for Cloud Management Platforms visit www.morpheusdata.com/gartner-cmp-mq.
- We'd love to learn more about your journey and discuss how we can help you along the way. Reach out to us at www.morpheusdata.com/demo.