

Why Your DevOps Best Practices Won't Work on Legacy Infrastructure

Research has found that private clouds beat public cloud costs at a certain scale[1]. Forrester research shows that private clouds are growing rapidly in regulated industries[2]. Companies like ZeroStack that provide private cloud using cloud-based management and operations show 50% cost savings over AWS even at a scale of just 10 physical servers[3].

Application development teams are constantly under the gun to accelerate the software delivery process, driven by time-to-market and competitive pressures. However, these teams often run into frustrating delays getting access to scalable and resilient infrastructure. On the other hand, IT and operations teams -- who are often responsible for delivering and managing this infrastructure--have been unable to provide capacity on demand to satisfy the rapidly-evolving agility requirements of application developers. IT and operations teams faced their own challenges of limited resources and unpredictable requirements from dev/test, lack of automation, and lengthy procurement and provisioning cycles, all resulting in over-purchasing and under-utilization.

Frustrated by this lack of agility, a growing number of development teams turned to the public cloud to accelerate their development release cycles. While this helped developers deliver applications faster, it also bypassed their company's security and governance policies, leading to the much-maligned "shadow IT" problem. What's more, companies started realizing that using public cloud wasn't exactly cheap. They also recognized that they are getting "locked" into a proprietary public cloud stack.

A hybrid cloud which integrates a private cloud with one or more public clouds can provide faster application delivery and flexibility in choosing the best cloud for each application based on cost, utilization, performance and business goals.

However, building and operating private clouds has proven to be a significant challenge for most companies. The key component that pushes up the cost of private cloud is the operational complexity and the cost of experts needed to build and operate it. To lower that, you should look into managed cloud offerings that are not entirely handled by people but by smart software that drives automation and intelligence into the entire stack. Very few private cloud solutions in the market handle the full lifecycle (deployment, upgrades, monitoring, alerting, and on-going resource management) of the cloud automatically and also allow a seamless migration of workloads between clouds.

Why Your DevOps Best Practices Won't Work on Legacy Infrastructure

Private Cloud Checklist for Application Development and Operations/IT teams

Here are six essential elements that a private cloud solution should offer so developers can accelerate delivery timelines while IT/operation teams have a consistent automated environment they can easily manage without a lot of overhead.

Essential Elements of a Private Cloud

Self-service, API-driven infrastructure	<p>Infrastructure as Code - a key attribute to enable DevOps best practice. It allows developers to write code (which can be done using their favorite programming language) and make RESTful API calls to the underlying programmable infrastructure to manage initial deployments and configurations as well as manage ongoing automated dynamic provisioning of infrastructure, autoscaling, monitoring, and alerting.</p> <p>All this automation removes the confusion and error-prone manual steps for the entire application delivery process, including development, testing, staging, and production deployments. This in turn accelerates software delivery and increases quality.</p> <p>Additionally, if self-service is available as a SaaS-based delivery model, it is easy to add more features and workflows very quickly without having IT doing a major upgrade. If your solution comes with just an on-premise install, ask how much work IT has to do in order to maintain this environment and how often new features will get added.</p>
Built-in browsable categorized application store	<p>DevOps engineers need readily accessible Continuous Integration and Continuous Deployment (CI/CD) tools such as Jenkins, Git, Maven, JUnit, etc. to automate the development and test pipelines. Additionally, tools like Ansible, Puppet, and Chef help with automated configuration and lifecycle management of workloads. Application developers need to quickly integrate with and deploy middleware services like RabbitMQ, Redis or storage backends like MySQL, Postgres, Cassandra, or MongoDB. A single-click deployment of such services and other complex multi-tier networked tools and applications greatly automates, accelerates, and simplifies the development process</p>
Seamless migration between private cloud and public cloud	<p>Developers and testers often need much greater capacity than is available on-premises for scale testing, and they should be able to move their application to public cloud for such purposes. Alternatively, they may already have some applications running on a public cloud and they may want to bring them back to their private cloud for cost savings or better performance. For these use cases, a private cloud solution should offer seamless bi-directional migration between public cloud and the private cloud.</p>

Why Your DevOps Best Practices Won't Work on Legacy Infrastructure

Hyper-converged cloud: Eliminating silos in the infrastructure	The bane of operations people is siloed and disparate fiefdoms of servers, storage, networks, and security. This inevitably leads to complexity of operations, lack of consistency in the environment, and lack of automation which in turn increase costs and reduce response times to the application developers. A hyper-converged cloud design with a software-centric architecture tightly integrates compute, storage, networking and virtualization resources and other technologies from scratch in a commodity hardware box supported by a single vendor. This approach eliminates silos and lowers costs and complexity. It also makes it easy to start small and grow on demand while staying tightly right-sized on capacity and cost.
Dashboards & visibility for troubleshooting	One of the key advantages for a private cloud is that you can get complete visibility across the infrastructure and applications. In a public cloud, you can only get VM-level stats but have no control or visibility below that. Make sure that your private cloud solution can provide visibility to developers directly. Many vendors only provide access to IT, thereby controlling what developers can do and making IT a bottleneck in the application development and deployment process.
Built-in browsable categorized application store	DevOps engineers need readily accessible Continuous Integration and Continuous Deployment (CI/CD) tools such as Jenkins, Git, Maven, JUnit, etc. to automate the development and test pipelines. Additionally, tools like Ansible, Puppet, and Chef help with automated configuration and lifecycle management of workloads. Application developers need to quickly integrate with and deploy middleware services like RabbitMQ, Redis or storage backends like MySQL, Postgres, Cassandra, or MongoDB. A single-click deployment of such services and other complex multi-tier networked tools and applications greatly automates, accelerates, and simplifies the development process
Better insights to improve efficiency and capacity management	A cloud requires capacity planning, utilization monitoring, right-sizing of workloads, and detecting zombie VMs and unused resources. Look for a solution that comes with built in analytics, management tools and insights. An even better approach would be to have a solution where you don't have to install the management tool on-premise and it can deliver its value as a service.

Summary

To meet the needs of application developers and development frameworks and to create an efficient DevOps platform, companies should look at cloud architectures that leverage private clouds managed via smart software. This empowers application developers and operations teams to leverage a consumption model they want, while providing control to IT and making them relevant to your business success.

Integrated intelligence and insights delivered as a service ensure that you never have to use spreadsheets to decide what to purchase and when. Cloud solutions with smart software ensure that you never have to make a Level 1 support call again; your management software can do that for you.

Why Your DevOps Best Practices Won't Work on Legacy Infrastructure

ZeroStack offers a hyper-converged cloud product that address these six essential requirements and more for DevOps teams.

Checkout the ZeroStack architecture here: <https://www.zerostack.com/product/> and see how it can help with your DevOps automation requirements: <https://www.zerostack.com/use-cases/devops-automation/>

References

- [1] <https://451research.com/blog/779-451-research-finds-openstack-and-commercial-private-clouds-can-beat-public-cloud-on-cost-%E2%80%93-93-but-only-at-scale>
- [2] http://blogs.forrester.com/jennifer_adams/16-10-18-private_cloud_adoption_is_alive_and_well
- [3] <https://www.zerostack.com/how-to-cut-your-public-cloud-cost-by-50-percent/>
- [4] <http://www.thewhir.com/blog/moving-away-from-aws-cloud-dropbox-isnt-an-anomaly-and-heres-why>