



# HEAD IN THE CLOUDS:

Linux and Cloud-Native Architecture

Business moves faster than ever today. Building a world-changing application requires an operating system that can handle modern challenges and adapt to whatever's coming next. Cloud-native web applications are the future.

These large-scale, complex, apps demand speed, reliability, and flexibility from the software that undergirds them. Agile development methodologies arose as the human response to the need to innovate and respond quickly to market needs in a highly competitive market, allowing dev teams to come up with new ideas, build and test them fast, and iterate until they have a winning solution. However, this didn't solve everything: the rise of these methodologies coincided with the massive growth of servers needed to manage these web-based apps (as opposed to the outgoing generation of desktop-based applications) and the massive pressure to do everything fast—deploy servers faster, scale faster, deploy code faster. Information technology (IT) teams simply could not do everything fast enough.

Every development team needs three key abilities to succeed in today's high-speed, high-pressure software world:

1. Quickly create new servers—virtual, cloud, or other
2. Deploy software changes fast
3. Troubleshoot production application problems easily



# TECHNOLOGY IS THE KEY

---

The rise of large-scale, complex web apps requires a technological solution. Luckily, the cloud itself, Microservices Architecture, and containerization all empower the teams building the next generation of these applications to get the agility and reliability they demand, even as they scale.

## CLOUD

---

The cloud is everywhere, and that makes sense: it's cheap and it's easy to connect to the computing power and storage you need, on demand. It's no wonder that modern app developers are going cloud-native; it's the direction that computing is heading in! The cost of scaling vertically or horizontally by adding servers or resources is basically a non-issue today. The real differentiators for app builders are the speed and agility with which they can respond to customer and market needs.

The bottom line is, the organizations that can respond to market demands faster will thrive. The rest...less so. [MIT research](#) showed that Agile firms grow revenue 37 percent faster and generate 30 percent higher profits than non-agile companies. [PwC](#) reports that Agile projects are 28% more successful than traditional projects. And as if that's not enough to sway you, consulting giant McKinsey, in McKinsey Quarterly, Dec 2015 said that Agile organizations have a 70 percent chance of being in the top quartile of organizational health, the best indicator of long-term performance.

When heavy hitters like this are saying something like this over and over again, there's definitely something to it. And it makes sense. Business Agility drives value faster by increasing customer satisfaction, visibility, responsiveness, productivity, quality, employee engagement, innovation and revenue. There's really no downside. But that doesn't mean it's effortless to implement the systems that will make your organization agile. Achieving business agility is a process of weaving resiliency and adaptability into the fabric of your enterprise. And the cloud is a vital technological aspect of creating both.

First off, the computing-on-demand aspects of cloud systems and the fact that your org isn't directly responsible for the physical aspects of the servers themselves (the buildings, the hardware, the staff who tend to them, etc.), present huge cost savings. You don't have to staff and maintain tons of physical equipment, even when you're not using it; just pay for what you need, when you need it.



# GOING CLOUD-NATIVE

---

Just connected your app to a cloud system isn't an instant recipe for success. Traditional monolith applications can be deployed on the cloud and leverage some of the benefits, but this only goes so far. Only cloud-native apps can take full advantage of all that the cloud has to offer. The combination of microservices architecture, containers for deployment and packaging, DevOps processes, and continuous delivery and integration typically supported by a cloud infrastructure are key reasons why cloud-native development is superior to traditional processes. Cloud-Native computing dynamically orchestrate containers to optimize resource utilization. The benefits that cloud-native apps offer include accelerated time to market and value, cost optimization, and improved agility. Teams can deploy new ideas and make changes super-fast and end-users have a more robust experience, with less downtime and seamless access to the latest version of the app.

# MICROSERVICES ARCHITECTURE

---

Microservices are a software development technique—a variant of the service-oriented architecture (SOA) architectural style—that structures an application as a collection of loosely coupled services. In a microservices architecture, services are fine-grained and the protocols are lightweight. This improved modularity makes the application easier to understand, develop, test, and become more resilient to architecture erosion. It parallelizes development by enabling small autonomous teams to develop, deploy and scale their respective services independently. Microservices-based architecture that lets teams rapidly develop new features and choose the

most ideal technology stack for each microservice. It also allows the architecture of an individual service to emerge through continuous refactoring. Microservices-based architectures enable continuous delivery and deployment, which fits perfectly with the DevOps model and contributes to the agility of teams developing this new generation of apps.

# CONTAINERIZATION

---

Containerization allows teams to package software in such a way as to create lightweight services that can automagically (yes, it's a word!) scale up or down. The containers create an abstraction layer between the OS and process and an isolated environment to run with everything it needs. Their key strength is that when software is run in a container, the developer knows exactly how it will run—it's predictable, repeatable, and immutable. There are no unexpected errors when it's moved to a new machine or between environments. All of the application's code, libraries, and dependencies are packed together in the container as a unified artifact. Think of running a container like running a virtual machine, without the overhead of spinning up an entire operating system. Given the wide reach and variety of systems that modern web apps can expect to encounter, the freedom and reliability that containerization offers are self-evident.

# DEVOPS: BRINGING DEVELOPMENT AND IT TOGETHER

---

One unique methodology that is especially suited to cloud-native app development is called DevOps. DevOps is about improving velocity and quality by streamlining the collaboration between software developers and IT operations. The ultimate goal of this process is to automate the process of delivering software and making infrastructure changes. It's also a natural fit in the cloud. The barriers that normally exist in traditional data centers aren't an issue with cloud computing. PaaS and application hosting features make it really simple to deploy applications and make changes.

The adoption of DevOps has been driven by many factors, including the use of agile and other development processes and methods and the demand for faster production releases (from both developer and business unit stakeholders) and the wide availability of virtualized and cloud infrastructure (from internal and external providers). More and more development teams and companies are catching on to these trends and adopting internal processes that take advantage of them.

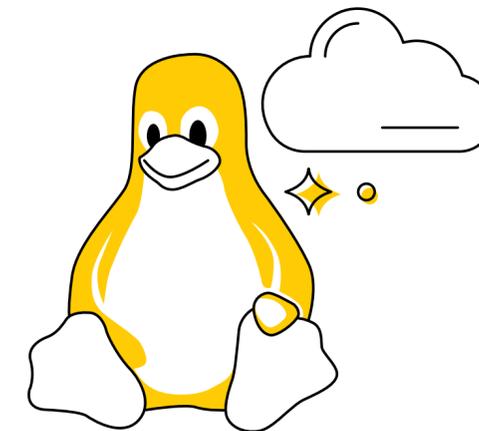
Under the previous software model, teams would build a product, then wait for specific times to deploy. It was a massive process where every element of the product had to be perfect before launch. Today, containers, microservices, and orchestration allow builders to keep building, iterating, and deploying on whatever schedule suits their needs, and the needs of their customers.

# LINUX AND THE CLOUD: A WINNING COMBINATION

---

Cloud-native architecture and DevOps processes are the keys to unlocking the potential of the numerous cloud services revolutionizing the computing world. They are also the key to rolling out higher-quality software, faster, and with more reliability. The final piece to this puzzle is picking the right operating system to tie it all together. Linux is choice OS for the cloud: as of 2017, Linux was running 90 percent of the public cloud workload. Every time someone uses Google, they're using a machine running the Linux kernel, not to mention the ubiquitous Android phone system, also built on Linux.

Companies and organizations the world over rely on Linux for critical workloads in data centers and cloud computing environments because of its high availability and reliability. Linux's design means it can support many use cases, devices, and target systems—whatever the workload demands. It's the natural technology for enabling cloud computing: modular, performant, power efficient, scalable, and open source.



# CONCLUSION

---

It's a cloud-based world. Building apps on a cloud-native architecture to integrate seamlessly with countless cloud services and environments makes sense from a business and a development perspective. Cloud-native architecture is perfectly suited for DevOps processes, by allowing containerization and incremental rollouts that allow for fast delivery, iteration, updates, and upgrades. And the OS most suited for this new paradigm: Linux. Every organization is struggling to find the best combination of software and processes to compete and thrive in the modern computing environment. If the cloud is your destination, then cloud-native architecture and Linux can get you there.

See how Sisense on Linux can  
supercharge your analytics offering

[Learn More](#)