9 Reasons Why Provisioning Tools aren’t ideal for Enterprise Application Deployment

Provisioning tools like Puppet, Chef, SaltStack and CFEngine are popular options in the configuration management space, but are they really suitable for enterprise application deployment? For many enterprise environments, application deployment is a bridge too far for system provisioning tools. Maybe it can be done, but it’s going to take a lot of scripting and workarounds to hook everything together.

If you’re about to adopt one of these tools you might want to stop and consider these factors first:

1. **Deployments must be orchestrated across multiple machines**
   - This holds for the majority of existing enterprise applications. It simply isn’t the case that an application is deployed to a single machine, or a set of identical “clones”, but system provisioning tools focus on that machine level.

2. **Most apps are not idempotent**
   - System provisioning tools tend to operate on the basis that if it doesn’t work the first time, you can just “run it again”. The problem is that most apps are not written that way. A typical app might, for example, create a schema the first time it is run, which won’t necessarily be removed if the deployment fails. How will the app handle it if the schema is already present when the deployment is retried?

3. **Platform and application may have separate lifecycles**
   - Is the development team really authorized to provide a full description of everything they require on the machines? What about legal and regulatory requirements, external development organizations, unforeseen support issues, or even simply a lack of expertise on the part of developers when it comes to properly configuring systems?

4. **Not designed for rapid application lifecycle**
   - At heart, system provisioning tools are designed to keep a system in a known versioned state. That’s an Operations approach, rather than a development approach. When you incorporate application deployment into your system provisioning, the application version is part of the system configuration. Creating new system configuration versions every time a new commit is made will quickly become unmanageable.
Agent-based
Relying on agents introduces unnecessary security risks and network chatter. There are inevitable system resource and maintenance overheads.

Limited Windows support
Many enterprises still run Windows rather than, or alongside, Linux. Some tools have reasonable Windows support, but that’s the exception. Most of the agentless system provisioning tools out there have no Windows support at all.

No concept of application, middleware, or environment
System provisioning tools set up files to copy and lists of commands to run on a box. The version of the overall machine configuration can be found, but how do you dig out which application version(s) are in there — a frequent question in an application delivery and management process.

Little or no out-of-the-box content for application deployment
You’ve got installation and configuration of the container, but what about the application layer? You may struggle to find content to handle this for WebSphere, JBoss and similar runtime environments.

Poor or no integration with build and CI tooling
How are your deployments and delivery pipelines triggered? System provisioning solutions’ poor integrations with build and CI tools mean that you have to find ways to “glue” the two together, which quickly gets messy. Addressing application layer challenges, and providing application-related functionality generally, simply aren’t a focus for these solutions.

If any of the above points are relevant to your applications or organization, consider an Application Release Automation tool for your deployments. Use system provisioning and configuration tools for what they’re really good at — getting your runtime platform to a known configuration state, and keeping it there — and manage your application tier changes using a deployment automation tool.

About XebiaLabs
XebiaLabs is the leading provider of delivery automation software that helps enterprises deliver higher quality software faster and more efficiently. XL Platform combines deployment automation, on-demand environment provisioning, Agile test management and enterprise release management for DevOps and Continuous Delivery. Using XL Platform, organizations reduce development applications costs, accelerate application time to market and bridge the gap between Development and Operations. Headquartered in the U.S., XebiaLabs has a world-wide network of sales offices and partners.

For more information visit: www.xebialabs.com
To request a trial visit: go.xebialabs.com/try-xl-platform