Agentless Architecture in a DevOps World

Introduction

There has been much debate over Agent vs. Agentless architectures for managing remote communication across distributed networks. Both of these architectures have been used in countless solutions, ranging from software application deployment and provisioning to data collection and system monitoring. While arguments can be made in support of both architectures, Flexagon believes that the Agentless architecture is the clear winner for deployment automation. We arrived at this key decision point after conducting thorough research and utilizing years of software design, development, and implementation experience.

An ***agent*** is a proprietary software program which is installed across distributed networks for the purpose of executing work and feeding data back to a centralized server.

In an ***agentless*** design, information is transmitted to or collected from computers without installing proprietary agents. This is accomplished by communicating with the software that is already installed on the computer, including the operating system and natively installed components. There are already more than enough native programs and protocols installed on a computer to establish the required communication without the need for agent software.

This paper describes the benefits of the agentless architecture, why it is the best fit for FlexDeploy and our customers, and how it positions us well for the future.

Simplified Management

Installing an agent on a remote computer generally involves logging into the host, installing the software, configuring the software, and starting a process. Sounds simple enough. Well, that is until you realize you have hundreds, or even thousands, of computers across your network. Not only do you need to perform the one-time installation for each agent, you also need to manage stopping, starting, and re-starting the agent process. And what happens when the next version or patch to the software requires an upgrade to the agent? The maintenance is annoying at best for smaller networks, and quite daunting for larger ones.

With an agentless architecture there is no process to manage. The connection is established by the centralized server to the target endpoints across the network using standard technologies and protocols that are available on those devices (e.g. servers, mobile devices, etc.) FlexDeploy utilizes SSH and SCP to connect and transfer information from the target endpoints.

Advocates of the agent-based architecture often point out that it offers greater reliability since agents running distributed work can continue operation during a network hiccup and report results back to the server at a later time. Point conceded for many applications of this architecture; however, factoring in the simplified maintenance and considering deployments are not generally considered transactional systems core to business operation, it's Flexagon's conclusion that the benefits of the agentless architecture far outweigh any reliability issues exposed during a network hiccup.

Trusted Security

The first version of the SSH protocol was released 20 years ago and has since been the de facto standard for securely accessing remote computers. Although it is certainly possible for an agent-based design to provide secured connectivity, SSH has been tried and true over all these years. Why not take advantage of it? While security architects and administrators are critical of agents and the potential security risks, SSH is generally accepted and no doubt already utilized across the vast majority of companies. SSH is well known, well documented, and battle tested. Quite simply, it works.

Another security consideration is that many agent-based solutions take the approach of requiring the agent to run either as root or as an administrator. This approach is often utilized to avoid the complexity of managing permissions required to accomplish the build and deployment tasks. Another approach has been to allow the user running the agent to impersonate another user using sudo rules. If impersonation is not supported, the alternative approach is to run one agent for each user that is required to accomplish its work. This can lead to doubling or tripling the number of agents to manage.

When agents are running on a production server as root, or any other privileged user for that matter, the server is left vulnerable to security attacks. Since agentless solutions do not require separate long-running processes on the target host, this security vulnerability is avoided all together.

Silence the Chatter

A key component to the agent-based design is that the agent is required to continuously ping the server to both indicate that it is running and to request more work. The amount of network chatter is not insignificant, especially after considering dozens, or even hundreds, of agents running across the entire network. This chatter can be throttled by reducing the ping time; however, this results in extending the idle time of the target hosts and increases the overall time to perform the core function – provisioning, configuration management, build, deploy, and testing.

With an agentless design the communication is established by the server to the appropriate target host only when there is work to be performed. No work, no needless chatter.

I Hear the Cloud a Comin'

Cloud computing has permeated the market over the last several years. According to Gartner, cloud computing will be the bulk of IT spend by 2016. Furthermore, more than 50% of U.S. businesses are now using cloud computing (whether public, private, or hybrid). The growth trajectory has been exponential, and there is no end in sight.

The Cloud presents an added challenge in that Software as a Service (SaaS) providers will never allow the installation of a third-party agent. While Infrastructure as a Service (IaaS) cloud would support installation of agents, it's a mixed bag with Platform as a Service (PaaS) providers. Given the range of support across cloud solutions and cloud providers, the agentless design is a clear winner in cloud computing.

It's worth mentioning that some SaaS providers will not only disallow installing proprietary agents, they will not allow remote access via SSH. In these cases, another computer must act a proxy and utilize APIs provided by the cloud provider to accomplish the deployment tasks.

Be ready for the Internet of Things

The software of today is no longer limited to running on servers in data centers. The ***Internet of Things*** is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices. These things include devices such as routers, smart phones, automobiles, machines and home appliances. According to Gartner, 26 billion devices will be on the Internet of Things by 2020. This is not limited to consumer facing devices either, as commercial applications can be referenced for every single industry.

These are devices where it is simply not practical and in many cases not possible to install proprietary agents. Connecting to devices using technologies and protocols which are part of the device is the path to the future. The Internet of Things is taking the world by storm. Be ready.

Conclusion

Like any architecture, careful consideration and detailed studies must be performed to ensure that it meets the needs of today and supports the advancing industry trends. Given the core benefits of the technology, security, and simplified maintenance, the agentless design is the top candidate over an agent-based design as it relates to DevOps tooling. Looking forward to the way of the future with Cloud Computing and the Internet of Things, Flexagon believes that the agentless architecture is no doubt the best fit.