Fast, Easy, and Automated Deployment of MicroStrategy Using SaltStack and GoCD

January 2018

Kyle Schustak
kyle.schustak@openx.com

Solomon Vedavanam
solomon.vedavanam@openx.com
Who are we?

**Kyle Schustak**
- Ex-MicroStrategy consultant and Tech Support Engineer (3 years)
- Specialize in MicroStrategy SDK & Server Administration
- Designing and building large-scale custom BI tools from the ground up

**Solomon Vedavanam**
- Design and development of large scale BI platform.
- Datawarehouse and ETL expert in designing scalable data systems.
- Microstrategy Architect, Administrator and SDK developer.
Who is OpenX?

At it’s simplest – we connect demand and supply

SUPPLY
(Publishers)

Programmatic Advertising Platform

DEMAND
(Advertisers)
Why quick, repeatable deployment is so important?

- We have multiple Intelligence Servers and multiple Web Servers and wanted to make sure each of them was identically configured.
- Assures that Development, QA, and Production environments have the same configurations.
- We occasionally need to spin up additional Web Servers quickly due to memory issues.
- We do every-other-week Web Server deployments to support our ever-evolving SDK code.
- Because the browser-side SDK code changes regularly and caching is important for performance reasons we needed a solution that supported cache busting.
- Allows for easy, seamless upgrade between MicroStrategy versions.
- Creates an environment for Continuous Integration and Continuous Delivery (CICD)
What technologies we used?

SaltStack

- Salt is open source software for IT automation.
- Capable of providing command and control of massive, complex technology implementations, Salt is also used for small, simple use cases.
- Salt is an intelligent, powerful and flexible open source software for remote execution, configuration automation, cloud control and event-driven orchestration of the most challenging DevOps and IT ops tasks.
- We utilize Salt to make sure that packages and files are deployed on specific machines at the correct locations.
- Salt also executes custom commands and assures that the Web Servers are connected to the Intelligence Servers
- For more info see: https://saltstack.com/
What technologies we used?

GoCD

- GoCD is an open source tool used to automate and streamline build-test-release cycle for reliable, continuous delivery of your product.
- GoCD creates and uses easy visualizations to see and compare the status of releases.
- GoCD is crucial for our continuous integration and continuous delivery system, but is not involved in our initial installation process.
- For more info see: https://www.gocd.org
What technologies we used?

Apache Maven

- Apache Maven is a software project management and comprehension tool.
- Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.
- We use Maven to package our custom SDK code into a war file that is used for deployment.
- Maven also assists in our custom mechanisms for cache busting and running unit tests.
- For more info see: https://maven.apache.org/
What technologies we used?

Apache Ant

- Apache Ant is a Java library and command-line tool used to drive processes described in build files as targets and extension points dependent upon each other.
- The main known usage of Ant is the build of Java applications, but can be used to build other application types, such as C or C++.
- We use Ant to build our java code and assure that only the desired files are included in our finalized war file.
- Ant also plays a key role in our cache busting mechanism and handles minimization of JavaScript and CSS files to improve end user performance.
- For more info see: [http://ant.apache.org/](http://ant.apache.org/)
What technologies we used?

- GitHub
  - Source control

- JFrog Artifactory
  - Artifact repository
How did we combine these technologies?

Developer commits change to GitHub
How did we combine these technologies?

GoCD sees the change and begins build.
How did we combine these technologies?

GoCD uses Maven and Ant to build the war file.
How did we combine these technologies?

GoCD uploads the war file to Artifactory
How did we combine these technologies?

GoCD triggers a Salt deployment of the war file.

JFrog Artifactory
How did we combine these technologies?

Highstate is triggered on the server.
How did we combine these technologies?

Highstate sees there is a new war file available and downloads it.
How did we combine these technologies?

Salt configuration is used to place the war file in the correct location and connect W-Server to I-Server(s).
How did we combine these technologies?

Developer commits change to GitHub

GoCD sees the change and begins build

GoCD uses Maven and Ant to build the war file

GoCD triggers a Salt deployment of the war file

GoCD uploads the war file to Artifactory

Salt configuration is used to place the war file in the correct location and connect W-Server to I-Server(s)

Highstate is triggered on the server

Highstate sees there is a new war file available and downloads it
How did we combine these technologies?
Questions?