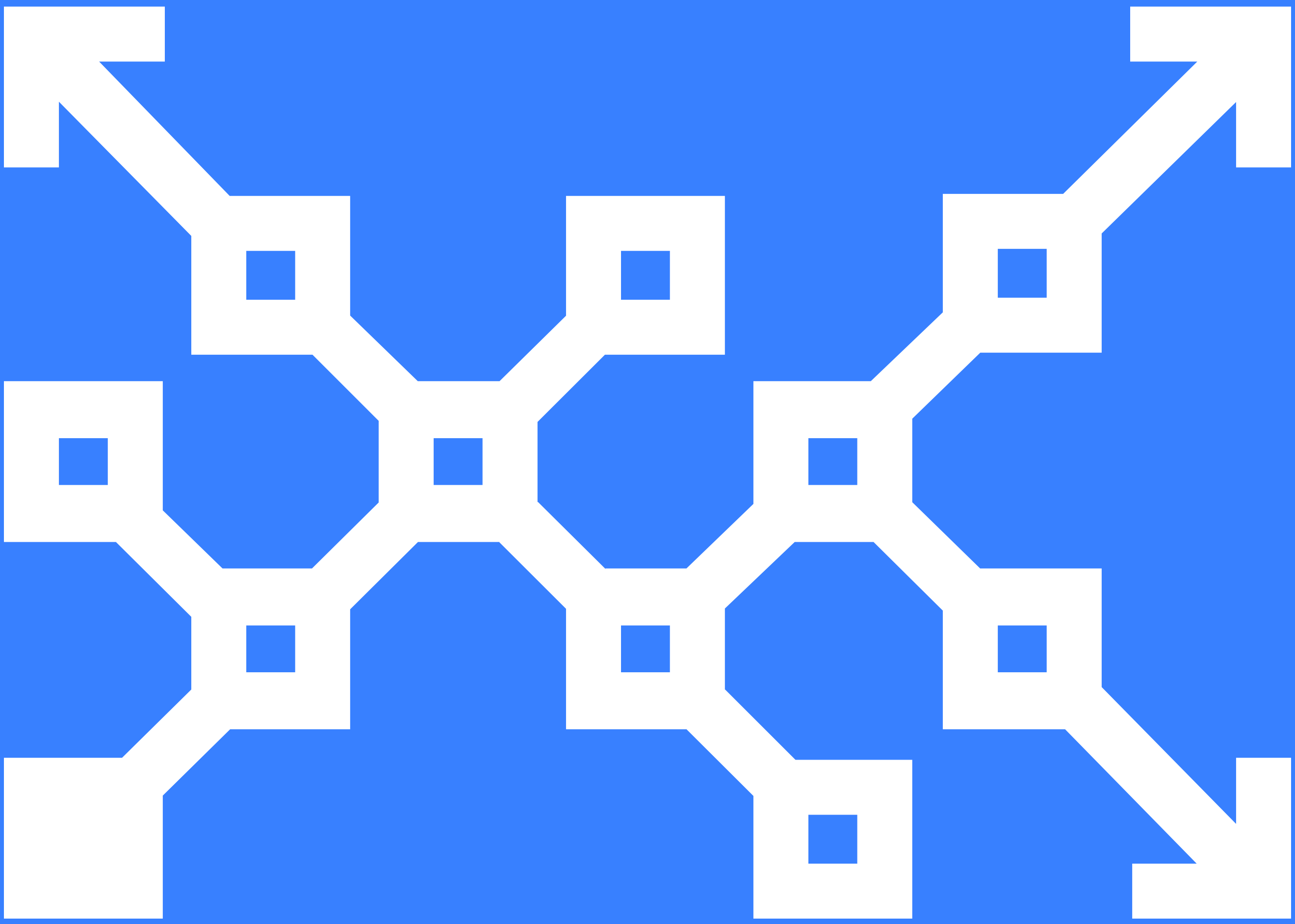
WHITE PAPER

Puppet Bolt and Splunk

Moving from reactive Ops to proactive automation



Welcome to the new era of DevOps. Right now, organizations of all sizes are in the midst of transformation. They are reducing complexity, standardizing their tools, and collaborating to accelerate and scale DevOps across their teams and initiatives and then they're running into a decidedly old-school problem: manual troubleshooting.

IT staff are still spending hours every day reacting to incidents and remediating issues manually. In fact, according to a recent industry survey, troubleshooting is a daily frustration for many tech managers, including more than half (53%) of DevOps practitioners.

As a result, automated infrastructure delivery and self-service environments—the final stages of the DevOps journey — remain out of reach for countless teams. As long as day-to-day emergencies continue to arise, Ops remains too busy firefighting to even consider which activities to automate.

If these issues sound familiar to you, the promises of today's DevOps opportunities are closer — much closer — than you may think. Troubleshooting may seem like a distraction, but it is in those incidents that your map toward automation will emerge. And if you use Splunk, you are already well-equipped for the journey ahead. Splunk can give your team visibility you need to uncover real opportunities for automation. All you need next is Bolt — and an open mind.

What is Bolt?

Bolt is an open source, multi-platform tool that allows any organization to start automating — no prerequisites or Puppet knowledge required. Bolt gives users the power and flexibility to run ad hoc commands, scripts, tasks, and plans across infrastructures and applications.

Bolt is agentless. Through a command line interface (CLI), a Bolt user can connect remotely to devices via SSH or WinRM and execute commands in any language, on any platform.

The tool makes it easy for operations to perform actions like:

- Starting and stopping services
- Rebooting remote systems
- Customizing virtual machines
- Patching security vulnerabilities
- Deploying apps
- Gathering package information and system facts

For many users, the path to automation in Bolt begins with the conversion of scripts they are already running manually into **tasks** they can fire off on demand. Add a metadata file to a script, and that script becomes a documented, repeatable, and shareable task. Each task gets uploaded to the Puppet Forge, where it can be shared with everyone else at your organization. There, you have automated.

When tasks need to happen in a certain order, you can use Bolt to chain them into workflows called **plans**. Like shell scripts, plans ensure consistent, sophisticated orchestration across your infrastructure. These and other Bolt features were created to help IT teams quickly troubleshoot and deploy point-in-time changes to systems.

But automation necessitates more than speedy incident response times; it demands critical thinking and strategic planning. Moreover, it would be a mistake to try to automate anything and everything without the requisite perspective and experience. Teams need to consider the scope, frequency, and complexity of each problem. Without actual insights, this is nothing more than a theoretical — and theoretically infinite — "what-if" exercise.

To harness Bolt's true potential and start automating, your team needs actionable visibility into the delivery lifecycle. In other words, you need to know what's broken before you can fix it.

To harness Bolt's true potential and start automating, your team needs actionable visibility into the delivery lifecycle. In other words, you need to know what's broken before you can fix it.

Enter Splunk

Splunk provides teams with valuable real-time data insights and visualizations. If your team uses Splunk, you may already know how the software's dashboard, reporting, and predictive analytics features can fuel innovation at your organization.

If your team hasn't yet applied Splunk to DevOps, however, you could be missing out on the software's greatest benefits. Splunk enables teams to spot and address obstacles and errors throughout the app lifecycle, from coding to staging, to testing, to deployment.

This isn't only a matter of optimization. Splunk allows you to detect patterns in your development and operational approaches. Over time, you can see where your team spends a surprising amount of energy responding to repetitive incidents that are relatively easy but tedious nonetheless. This level of deep, long-term visibility makes Splunk the key to seeing what, where, when, and how you can automate.

Ready to see how it all fits together?

How to drive automation with Bolt and Splunk — in 4 steps

1. Find an issue, fix an issue

Imagine your team has discovered an issue through Splunk. Your web servers are running out of memory. Without Bolt, the only way to fix the issue is to log in and reset each machine by hand.

With Bolt, your team can meet the demand quickly and efficiently from a single device. Better yet, they can use one of their existing scripts or shell commands to do so. The following Bolt plan demonstrates one approach:

```
bolt task run service name=httpd action=restart --targets
web0.example.com,web1.example.com
```

or

```
bolt task run reboot --targets
web4.example.com,web5.example.com
```

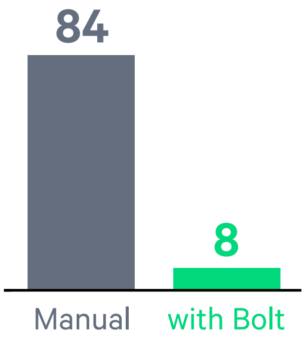
(Note that a single command restores servers with this known problem. Use one of the following, not both, depending on how your organization handles such issues)

2. Feed more data to Splunk, gain deeper awareness

When you want to see the impact of a Bolt command or plan, it's easy to report the results to Splunk. With the splunk_hec module, you can use Bolt to submit data directly to Splunk for viewing and analysis.

In the chart shown here, we can see how much time we save by running a Bolt plan rather than manually logging into and resetting each machine.

As your Bolt usage scales, so will your awareness and visibility into your organizational data. Splunk is designed to process virtually any data type and provide insight in any environment. The more hosts on which you run commands, and the more commands you run, the more valuable your analytics in Splunk.



Time (in seconds) to restart services on 5 web servers

LOWER IS BETTER

3. Power up Bolt for scale

After feeding more data into Splunk, let’s say your team has discovered another issue: a number of machines in your fleet are failing to synchronize with the NTP server. Your team doesn’t yet know why, but they know how to solve the issue: by logging into and resetting each device.

If only they could declare that desired end state once and apply it everywhere. With Bolt, they can. A simple feature gives Bolt users the same powerful outcomes DevOps practitioners can achieve with Puppet.

Bolt **Apply** lets your team use Puppet code in their Bolt plans, so the team can selectively check or enforce a state across your infrastructure without a Puppet master server.

The following code demonstrates how your team can use Bolt Apply to check for NTP and configure it consistently across your fleet — without manually logging into a host. See the sidebar for steps to test this code.

```
plan example::ntp(
  TargetSpec $targets
) {
  apply_prep($targets)
  apply($targets,,) {
    class { ‘ntp’:
      servers => [ ‘ntp1.example.com’, ‘ntp2.example.com’ ],
    }
}
```

Bolt Apply allows us to run a plan that inspects device configurations by transparently leveraging features found in Puppet and safely updates any machine that is not consistent with our predetermined setting.

To run this code do the following:

- Create a new Bolt Project:
  `bolt project init --modules puppetlabs-ntp ~/example`
- Create new example plans directory:
  `mkdir -p ~/example/site-modules/example/plans`
- Place example code in the following file:
  `~/example/site-modules/example/plans/ntp.pp`
- Run command:
  `bolt plan run example::ntp --targets web4.example.com,web5.example.com`

Directory contents will look similar to the structure in the sidebar.

```
example/
├── bolt.yaml
├── modules
    └── ...
            └── ...
├── Puppetfile
└── site-modules
    └── example
        └── plans
            └── ntp.pp
```

4. Automate

When it’s time to expand our Bolt plan and apply it to NTP settings on 500 machines, you will notice one thing right away: it’s the exact same code for working with one machine. The same code operates in the same way on one, two, 500, or 5,000 nodes!

When it’s time to expand your Bolt plan, you’ll notice one thing right away— the same code operates in the same way on one, two, 500 or 5,000 nodes!

This is where automation becomes a real time-saver. As we scale up the number of servers for our Bolt plan, it’s not going to take us that much longer. Time taken for manual tasks scales linearly as the number of servers increases. In other words, if you have to manually restart services on 100 servers, you’re looking at upwards of an hour spent running commands individually. Bolt has the advantage of running tasks in parallel, automatically, which results in exponentially more servers needing much less time. This turns an onerous and repetitive manual chore into an opportunity to make a quick visit to the coffee machine.

This is one example of the value of an approach that uses Splunk and Bolt to collect data, identify common tasks and hosts that need intervention, and remediate issues reactively. But what if you could manage it all before the need for troubleshooting even arose? With Bolt and Puppet Enterprise, you can. The same workflows you run in Bolt can be used in Puppet Enterprise as Puppet Tasks.

In our example, the team resolved the NTP issue, but they would need to follow the procedures all over again if and when the problem re-emerges. In other words, they’d continue reacting. By moving their workflows from Bolt to Puppet Enterprise, your team can automate the response, applying the fix in advance of an incident or outage. That’s what proactive automation looks like.

Find issues with Splunk, fix them with Bolt, and then scale and automate the solution throughout your organization with Puppet Enterprise. Welcome to your new era of DevOps.

See how easy it is to go from reactive troubleshooting to proactive automation. Get started with Bolt**.**