This document will go over what I do when I demonstrate smart management (sm) and why I do those things.

First off, why should a customer care about using sm? For any customer with more than ten systems, it is easy to get something wrong with patching. They could have an error with pulling down packages (network, repodata, dep solving, …), they might simply forget or accidentally skip a system, they might install the wrong version of a package (installing an untested version in production), or something else. There are lots of 3rd party solutions for this such as big fix, manage engine, bmc, managing your own repo server, or something else, so why should a customer use satellite?
IDC: Red Hat Satellite Helps Enterprise Organizations Optimize Infrastructure with Automation Tools ---- 18 pages
IDC: infographic - business value of Satellite --- 1 page
--- Email pgervase@redhat.com for those documents.

Basically

- Official Red Hat product, so fully supported by Red Hat
- Integrates with Insights
- Easy patch management (content views, actual patching), reporting
- Cost effective (refer: value docs)

OK, on to the actual demonstration!
Here is a link to my slides:
https://people.redhat.com/pgervase/presentations/how.to.demonstrate/smart.management/opendemo.pdf
Feel free to email me at pgervase@redhat.com if you want the source for the slides.

Here is a link to one of my open demos:
https://bluejeans.com/s/x_F8WXsu0VQ

The open demo takes about 50-55 minutes, but if you knew the audience wanted to go in depth in some topic such as the api, provisioning, patch management, or something else in addition to the general overview, it could easily go longer. I start off rushing through the slides so that I can get to the actual demonstration. For the slides, I do a quick overview of what Satellite / Smart Management can do (content management, patching, provisioning) and then a few example deployments using the 50 satellite infrastructure subscriptions. Make sure you mention that those 50 sat infra subs are for just satellites and capsules, not regular clients. I mention this again when discussing activation keys later on.

For explaining content views, I like to use a restaurant menu analogy - you walk in to the restaurant and over to the side is the bar, so you sit down there and the bar menu has 50 or 100 drinks or whatever. If you tell them that you want to order off the dinner menu, they'd likely tell you to go to the main restaurant for that. You sit down in the main restaurant and you see the kids menu. OOH! Kids mac and cheese with dinosaur nuggets! You try to order that and they'll tell you that you're an adult and you must order off of the adult menu. The content views, then, are doing basically the same thing - they're listing which packages (menu items) are available to be installed on which systems. You don't *need* to install every dessert from the dessert menu, but those are the ones that are available to you. The various systems don't need every package from their content view installed, but those are the packages available to them.

For content view versions, I take the menu analogy one step further. The fancy restaurants have seasonal menus. You were at the restaurant a few months ago and they had a fresh raspberry dish. Now that it's winter time and they have the winter version of their menu, that dish is not listed and so you are not able to order that dish. For content views, different versions of the same content view will have different packages and different versions of the packages from earlier versions.

After all that, I probably do some demonstrations - making a content view, changing the version, promoting the content view. For this, I use the satellite tools repo since it's so small that it doesn't take long for any of those tasks.

I then do a demonstration of registering a client using user / pass and then show using activation keys. For this, I have two vms, {rhel7dev,rhel7prod}.usersys.redhat.com so you can show how the "prod" system can register with the activation key and get the prod content view, the correct subscription (here is where I again mention the 50 sat infra subs and how using an activation key ensures that you're not accidentally consuming one of them), correct repos, host collection, … set by the activation key.

Then do some compute resource, host group, … demonstration. I have a vmware hypervisor and before that I used a kvm hypervisor, so I'd show all the vms in there, even those that aren't registered to the satellite. This is how you setup provisioning. I show compute profiles and might make a new one. If I'm moving quickly enough through the demonstration, I might show how to provision, but I usually skip over that and simply talk about it.

You can show how to setup authentication such as ldap, but I likely skip over that.

Do some Red Hat Insights demos - easy demos are fixing the smbloris vulnerability and fixing a mangled /etc/sysconfig/selinux file. Neither of those are out of date errata issues and the system can keep working, but insights still flags them before you have an issue -- preemptive remediation, proactive not reactive, ...

Finally, show some api stuff if you have time. Here are some example scripts:
https://people.redhat.com/pgervase/sat6/
There is an apidoc in the satellite:
https://satellite.example.com/apidoc/v2

Here is the api guide from the product docs:

https://access.redhat.com/documentation/en-us/red_hat_satellite/6.9/html-single/api_guide/index

Please email me if you would like access to my satellites or capsules.

Please let me know if you have any questions!!
pgervase@redhat.com