Stephen Gallagher's Open-Source Blog

Keeping the undesirables out of your system

Proposal: FreeIPA Role for Fedora Servers Using Cockpit

Posted on 2013/12/09 | 2 Comments

At last week's Fedora Server Working Group meeting, we encountered some confusion as to what exactly a role should look like. We seemed to be of differing opinions, technologically, on how to implement the roles. I recommended that we take a step back and try to design the user experience that a role should accomplish first. I recommended that we probably wanted to take one example and provide a user story for it and use that as a straw-man to work out the more general cases of server roles.

What is FreeIPA?

FreeIPA is a comprehensive identity-management and domain controller system, providing LDAP and Kerberos services.

How would I get it up and running?

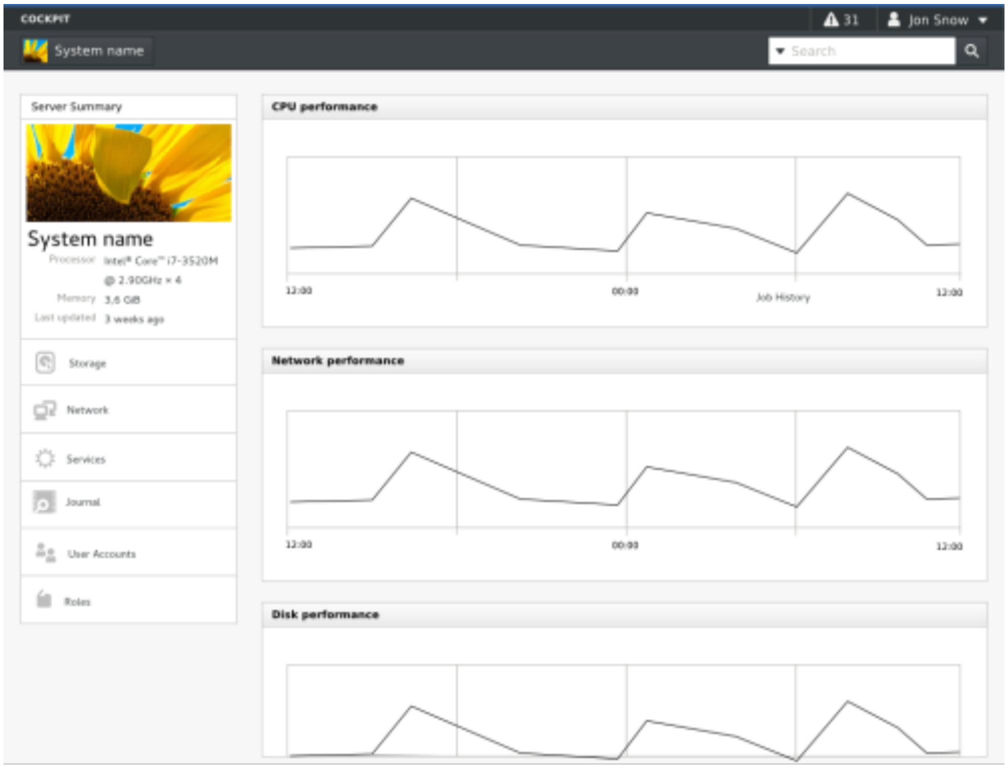
TRADITIONAL APPROACH

In the world of Fedora 20 and earlier, the expected behavior would be that the user would first install the appropriate set of packages (either as part of installation using the GUI or kickstart or by installing the packages manually using yum on a running system). They would then run a command-line tool (ipa-server-install) to perform the configuration and initial setup of the domain (being prompted with a lot of low-level information). Joining other machines to that new domain would then be done manually or via scripts using other command-line tools.

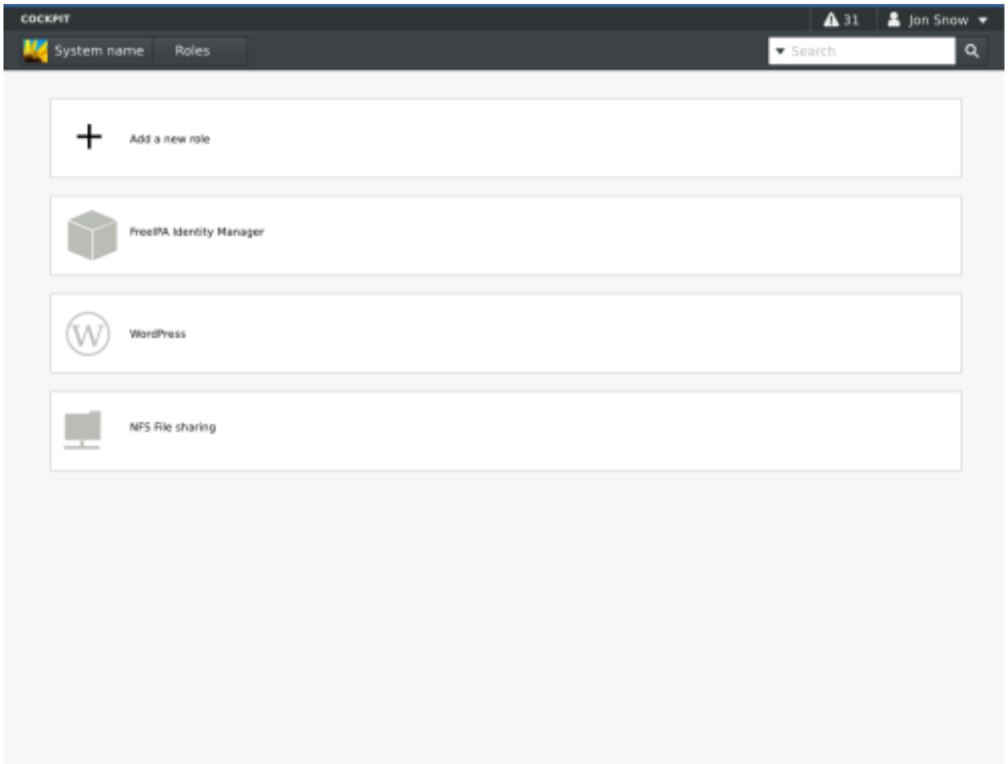
The process to enroll additional FreeIPA domain controllers (aka replicas) is very similar: install all of the patches and then run ipa-replica-install, providing the necessary configuration information in an answer file or interactive prompt.

ROLE-BASED APPROACH

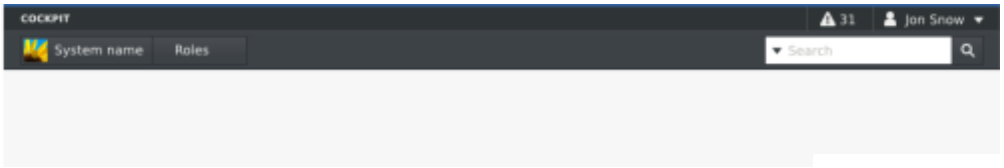
The goal of the role-based approach would be to present the minimal amount of information to the user necessary to produce a working FreeIPA domain. Additionally, it should be easy to determine from a management console which machines are configured in this manner. For this proposal, I am going to use a hypothetical Cockpit role interface to describe and illustrate the process. Please note that these are mock-ups and not representative of a final design.



Dashboard for individual server, roles at bottom



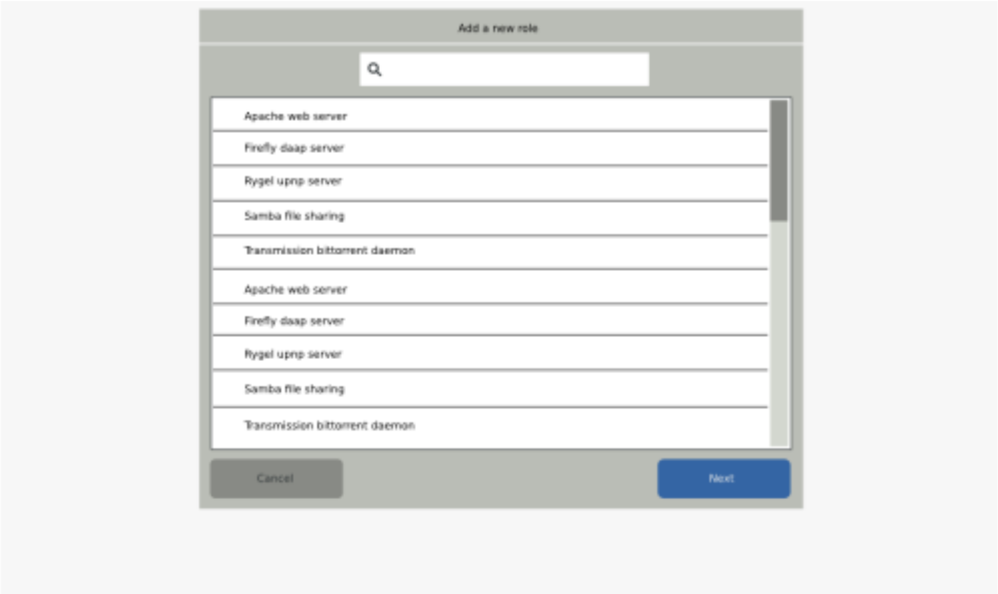
In Roles you see existing roles



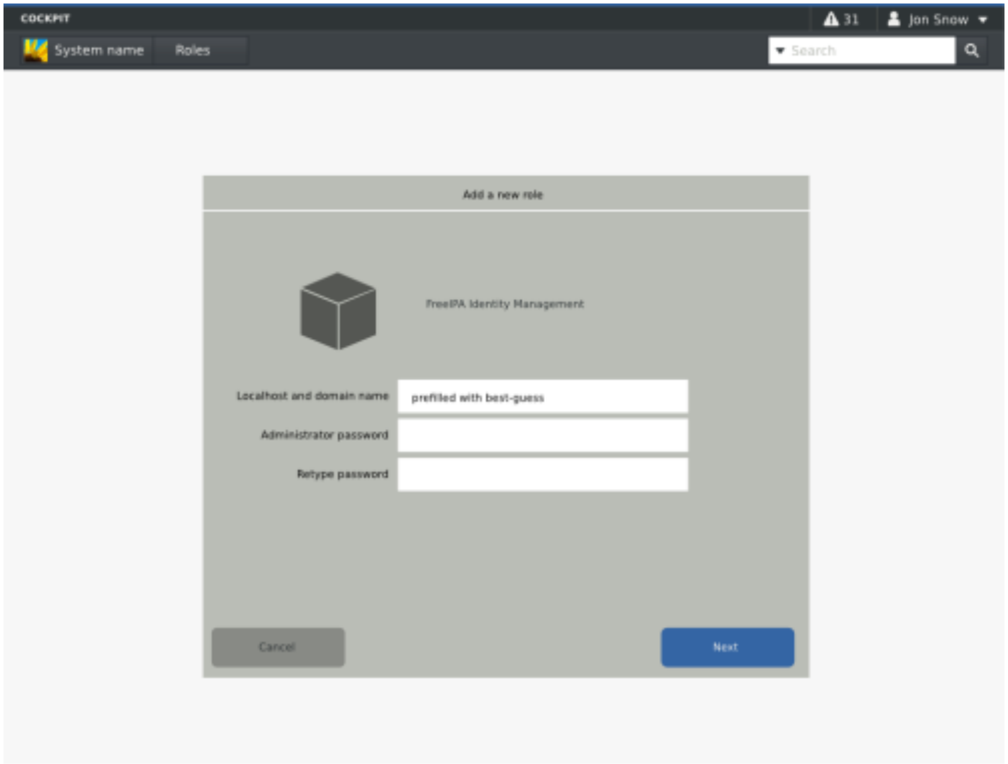
Once we connect to the machine, we would first want to examine the roles currently assigned to that machine. Presuming that the FreeIPA/Domain Controller role is not already on the machine, we would then elect to deploy one there. (We’ll assume this is the first such machine now and defer discussion of replicas).

From here, we select the FreeIPA Domain Controller and proceed to query the user for the minimal set of information necessary to deploy a FreeIPA server.

Here, the minimal set of information is the domain name (which may be auto-guessed from the machine’s `hostname -d` as well).



Pops up dialog with a list of possible roles

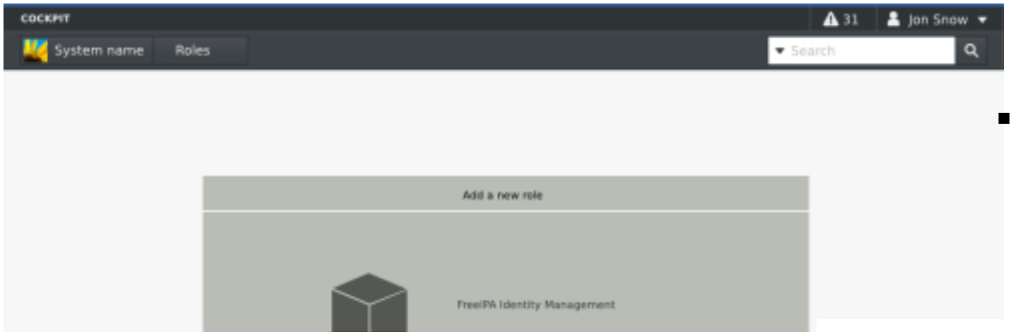


Localhost and domain name: prefilled with best-guess

Administrator password:

Retype password:

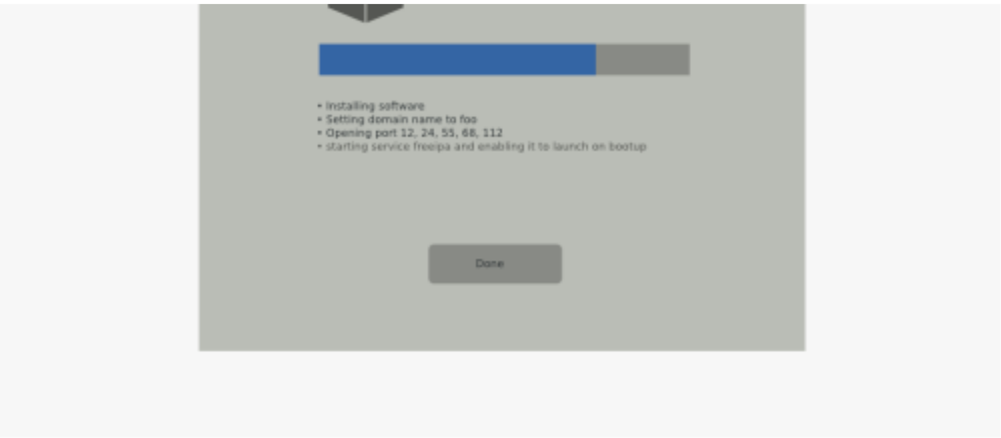
Ask all the neccesary questions



The next step here shows the majority of the operations representative of my view of a role. It's not enough to install the necessary packages and then leave things to the administrator to complete (or script) the setup. Up until this point, we are doing the information-gathering for a role deployment.

Here, we demonstrate the actual deployment of a role. At this point, Cockpit (or whatever other management tool we are using) should be relaying the gathered data to the managed system and the role-installation mechanism should take over. In the FreeIPA case, this means:

- Package Installation

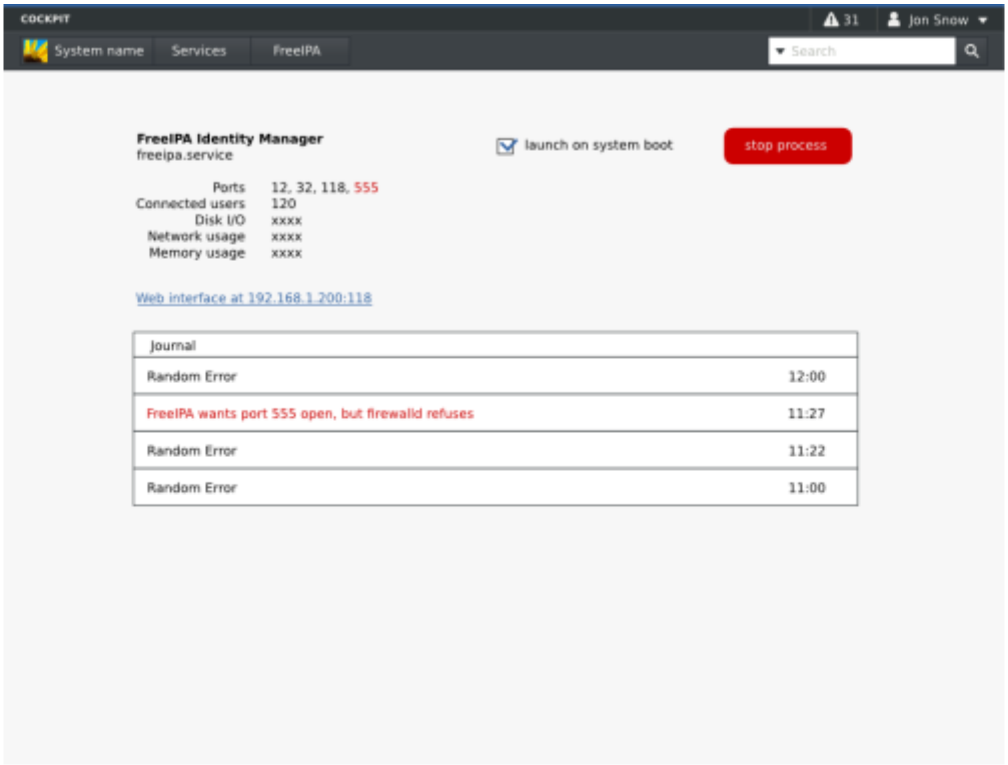


- Installing software
- Setting domain name to foo
- Opening port 12, 24, 55, 68, 112
- starting service freeipa and enabling it to launch on bootup

Installs. Cockpit should have all the data needed for FreeIPA to consume

- Hostname/Domain configuration
- Installation and configuration of the FreeIPA server (unattended execution of ipa-server-install)
- Opening of the appropriate firewall ports
- Configuring the services to start immediately and persistently

Once installation is complete, we should be able to view the status of the FreeIPA services through either the role or services tabs in Cockpit. This display should also include a link to the FreeIPA Web UI for full management.



FreeIPA Identity Manager
freeipa.service

☑ launch on system boot

Ports	12, 32, 118, 555
Connected users	120
Disk I/O	xxxx
Network usage	xxxx
Memory usage	xxxx

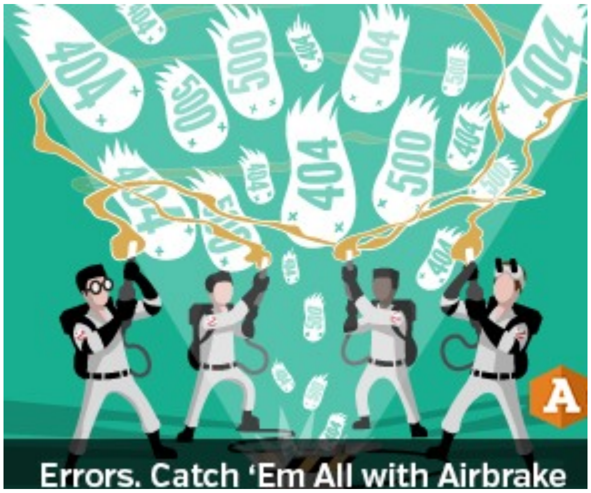
Web interface at 192.168.1.200:118

Journal	
Random Error	12:00
FreeIPA wants port 555 open, but firewalld refuses	11:27
Random Error	11:22
Random Error	11:00

Under the inidvidual service page is a url to the FreeIPA web UI.

Additionally, the role interface should also offer simplified deployment of descendant roles, such as adding DNS services to a FreeIPA domain, but that's not currently represented in the mock-ups (due to time-constraints).

All images above are copyright "The Cockpit Project" and licensed CC-BY-SA 3.0.



Share this: Twitter 2 Facebook 1

Like this: ★ Like

Be the first to like this.

Related

Flock 2013 Con Report	One Week With GNOME 3 Clas…	IETF, MTI codecs and Fedora
In "Fedora"	In "Fedora"	In "Fedora"

This entry was posted in Cockpit, Fedora, opensource, Server. Bookmark the permalink.

2 RESPONSES TO PROPOSAL: FREEIPA ROLE FOR FEDORA SERVERS USING COCKPIT

Adam Young | 2013/12/09 at 3:26 PM | Reply

Roles are explicitly for managing ACLs on the Directory server. We've looked in to expanding the use of roles beyond that, and there are a couple of reasons to resist.

First is that there is a lack of standards of what roles should look like. I think that is what you are trying to address in this article.

Second, and probably as important, is that a Role is only for cross cutting concerns in the Identity Management domain, so for HBAC etc, yes. But not for roles in an application deployed on top of an IPA based system. The term I've used in the past is "centralized authentication, local authorization" for that.

Follow

Note that I wrote up several blog posts about delegation and roles in FreeIPA a few years ago that describe how to manage a subordinate grouping, whether for users, hosts, or DNS:

http://adam.younglogic.com/2012/02/group-managers-in-freeipa/
http://adam.younglogic.com/2012/02/hostgroup-managers-freeipa/
http://adam.younglogic.com/2012/02/netgroup-managers-in-freeipa/
http://adam.younglogic.com/2012/02/dns-managers-in-freeipa/

I would suggest that there be two roles to support the above use cases:

1. host-enroller: this would be a minimal role used only for enrolling hosts that do not have an OTP and host entry in FreeIPA prior to enrolling. It would bascially put a host in "enrolled, least priviledges" mode.

2. replicator: for setting up replication agreements.

Deploying a FreeIPA server obviously is going to predate any roles enforced by the IPA server. I am assuming that you are talking about using the term Roles for an application that is then going to deploy FreeIPA for you. "The goal of the role-based approach would be to present the minimal amount of information to the user necessary to produce a working FreeIPA domain."

A role is typically a grouping of finer grained permissions. The idea is that by granting someone a role, you are granting them all of the permissions they need to get a specific job done.

> **sgallagh** | 2013/12/09 at 3:32 PM | Reply
>
>
>
> I think I may have been unclear in my explanation above. Unfortunately, there's an overlap on the term "roles". I was actually talking about roles of a MACHINE, not roles interacting with FreeIPA. ("Role" is almost as bad as "domain" for overuse).
>
> I'm talking about server roles. So things like saying "That (physical or virtual) server over there is now a FreeIPA domain controller" and "That one over there is a PostgreSQL database server". Basically, package installation plus minimum configuration to make it useful (if not efficient).