Subject: Re: Re: [RFC] [PATCH 0/7] Some basic vserver infrastructure
Posted by dev on Wed, 29 Mar 2006 12:07:21 GMT
View Forum Message <> Reply to Message

Serge,

Serge E. Hallyn wrote:
> Quoting Kirill Korotaev (dev@sw.ru):
>> Just to make it more clear: my understanding of word "nested" means that
>> if you have, for example, a nested IPC namespace, than parent can see
>> all the resources (sems, shms, ...) of it's children and have some
>> private, while children see only its own set of private resources. But
>> it doesn't look like you are going to implement anything like this.
>> So what is nesting then? Ability to create namespace? To delegate it
>> some part of own resource limits?
>
> Nesting simply means that any child ns can create child namespaces of
> it's own.
your picture below doesn't show that containers have nested containers.
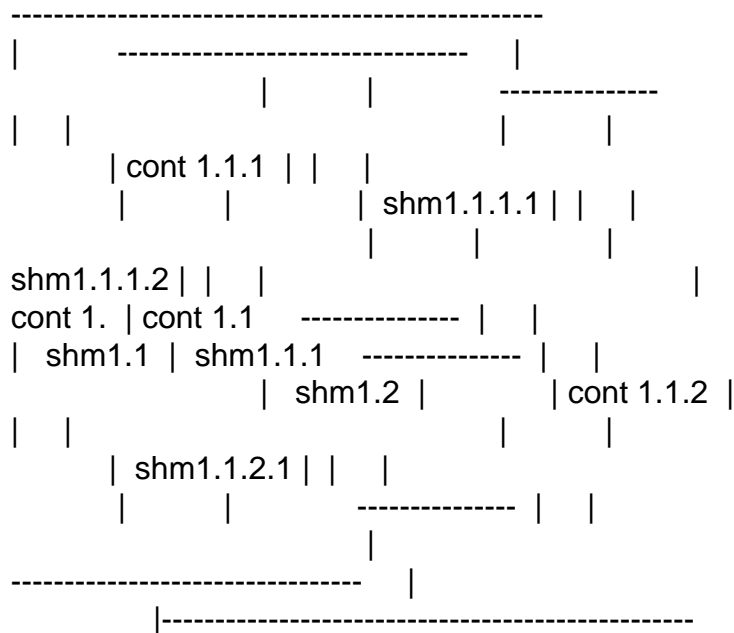You draw a plain container set inside vserv.
What I mean is that if some container user can create another container,
it DOES not mean it is nested. It is just about permitions to create
other containers. Nested containers in my POV is something different,
when you can see the resources of your container and your children. You see?

I will try to show what I mean on a picture:

```
--------------------------------------------------
|         ---------------------------------      |
                |       |        ---------------
|    |                             |       |
       | cont 1.1.1 |  |    |
          |       |          | shm1.1.1 |  |    |
                            |        |          |
shm1.1.1.2 |  |    |                            |
cont 1.  | cont 1.1     ---------------  |    |
|   shm1.1  | shm1.1.1    ---------------  |    |
                | shm1.2 |          | cont 1.1.2 |
|    |                             |       |
       | shm1.1.2.1 |  |    |
          |       |          ---------------  |    |
                            |
---------------------------------      |
          |------------------------------------------------
```

You see what I mean? In this example with IPC sharememory container 1
can see all the shm segments. while container1.1.2 can see only his
private one smm1.1.2.1.

And if resources are not nested like this, than it is a PLAIN container structure.

Kirill

> In particular, the following scenario should be perfectly valid:
>
> Machine 1   Machine 2
>   Xen VM1.1   Xen VM2.1
>   vserv 1.1.1   vserv2.1.1
>   cont1.1.1.1   cont2.1.1.1
>   cont1.1.1.2   cont2.1.1.2
>   cont1.1.1.n   cont2.1.1.n
>   vserv 1.1.2   vserv2.1.2
>   cont1.1.2.1   cont2.1.2.1
>   cont1.1.2.2   cont2.1.2.2
>   cont1.1.2.n   cont2.1.2.n
>   Xen VM1.2   Xen VM2.2
>   vserv 1.2.1   vserv2.2.1
>   cont1.2.1.1   cont2.2.1.1
>   cont1.2.1.2   cont2.2.1.2
>   cont1.2.1.n   cont2.2.1.n
>   vserv 1.2.2   vserv2.2.2
>   cont1.2.2.1   cont2.2.2.1
>   cont1.2.2.2   cont2.2.2.2
>   cont1.2.2.n   cont2.2.2.n
>
> where containers are used for each virtual server and each container,
> so that we can migrate entire VMs, entire virtual servers, or any
> container.
>
>>>>>> Perhaps we can get a ruling from core team on this one, as it's
>>>>>> aesthetics :-).
>> I propose to use "namespace" naming.
>> 1. This is already used in fs.
>> 2. This is what IMHO suites at least OpenVZ/Eric
>> 3. it has good acronym "ns".
>
> I agree.
>
> -serge
>