

---

Subject: Re: Re: [RFC] [PATCH 0/7] Some basic vserver infrastructure  
Posted by [serue](#) on Wed, 29 Mar 2006 13:47:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Kirill Korotaev (dev@sw.ru):

> 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.

And I am assuming that vserv is implemented as a container, hence this is an example of nested containers very likely to be used. But given what I now think is your definition of nested, I think we are agreed.

> What I mean is that if some container user can create another container,  
> it DOES not mean it is nested. It is just about permissions 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?

Alas, the spacing on the picture didn't quite work out :) I think that by nested containers, you mean overlapping nested containers. In your example, how are you suggesting that cont1 refers to items in container1.1.2's shmem? I assume, given your previous posts on openvz, that you want every shmem id in all namespaces "nested" under cont1 to be unique, and for cont1 to refer to any item in container1.1.2's namespace just as it would any of cont1's own shmem?

In that case I am not sure of the actual usefulness. Someone with different use for containers (you? :) will need to justify it. For me, pure isolation works just fine. Clearly it will be most useful if we want fine-grained administration, from parent namespaces, of the items in a child namespace.

-serge

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

```

>
> -----
> |           -----           |
> |           |           |           -----
> | |           |           |           |
> | | cont 1.1.1 | | |           |
> | |           |           | shm1.1.1.1 | | |
> | |           |           |           |           |
> 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 shmemory 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  
> >

---