Subject: [PATCH 0/6] preparations to enable netdevice notifiers inside a namespace (resend)
Posted by den on Thu, 31 Jan 2008 11:59:49 GMT
View Forum Message <> Reply to Message

Here are some preparations and cleanups to enable network device/inet address notifiers inside a namespace.

This set of patches has been originally sent last Friday. One cleanup patch from the original series is dropped as wrong, thanks to Daniel Lezcano.

Subject: [PATCH 1/6] [IPV4]: Fix memory leak on error path during FIB initialization.
Posted by den on Thu, 31 Jan 2008 12:00:45 GMT
View Forum Message <> Reply to Message

net->ipv4.fib_table_hash is not freed when fib4_rules_init failed. The problem
has been introduced by the following commit.
commit c8050bf6d84785a7edd2e81591e8f833231477e8
Author: Denis V. Lunev <den@openvz.org>
Date: Thu Jan 10 03:28:24 2008 -0800

Signed-off-by: Denis V. Lunev <den@openvz.org>
---
 net/ipv4/fib_frontend.c | 10 ++++++++-
 1 files changed, 9 insertions(+), 1 deletions(-)

diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d282618..d0507f4 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -975,6 +975,7 @@ static struct notifier_block fib_netdev_notifier = {

 static int __net_init ip_fib_net_init(struct net *net)
 {
+ int err;
  unsigned int i;

  net->ipv4.fib_table_hash = kzalloc(
@@ -985,7 +986,14 @@ static int __net_init ip_fib_net_init(struct net *net)
  for (i = 0; i < FIB_TABLE_HASHSZ; i++)
   INIT_HLIST_HEAD(&net->ipv4.fib_table_hash[i]);

- return fib4_rules_init(net);
+ err = fib4_rules_init(net);
+ if (err < 0)
+ goto fail;

```
+ return 0;
+
+fail:
+ kfree(net->ipv4.fib_table_hash);
+ return err;
 }

 static void __net_exit ip_fib_net_exit(struct net *net)
--
1.5.3.rc5
```

Subject: [PATCH 2/6] [IPV4]: Small style cleanup of the error path in rtm_to_ifaddr.
Posted by den on Thu, 31 Jan 2008 12:00:46 GMT
View Forum Message <> Reply to Message

Remove error code assignment inside brackets on failure. The code looks better
if the error is assigned before condition check. Also, the compiler treats this
better.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/devinet.c |   21 ++++++++----------
 1 files changed, 8 insertions(+), 13 deletions(-)

diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index 21f71bf..9da4c68 100644
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -492,39 +492,34 @@ static struct in_ifaddr *rtm_to_ifaddr(struct nlmsghdr *nlh)
  struct ifaddrmsg *ifm;
  struct net_device *dev;
  struct in_device *in_dev;
- int err = -EINVAL;
+ int err;

  err = nlmsg_parse(nlh, sizeof(*ifm), tb, IFA_MAX, ifa_ipv4_policy);
  if (err < 0)
   goto errout;

  ifm = nlmsg_data(nlh);
- if (ifm->ifa_prefixlen > 32 || tb[IFA_LOCAL] == NULL) {
-  err = -EINVAL;
+ err = -EINVAL;
+ if (ifm->ifa_prefixlen > 32 || tb[IFA_LOCAL] == NULL)
   goto errout;
- }
```

```
  dev = __dev_get_by_index(&init_net, ifm->ifa_index);
- if (dev == NULL) {
-  err = -ENODEV;
+ err = -ENODEV;
+ if (dev == NULL)
   goto errout;
- }

  in_dev = __in_dev_get_rtnl(dev);
- if (in_dev == NULL) {
-  err = -ENOBUFS;
+ err = -ENOBUFS;
+ if (in_dev == NULL)
   goto errout;
- }

  ifa = inet_alloc_ifa();
- if (ifa == NULL) {
+ if (ifa == NULL)
   /*
    * A potential indev allocation can be left alive, it stays
    * assigned to its device and is destroy with it.
    */
-  err = -ENOBUFS;
   goto errout;
- }

  ipv4_devconf_setall(in_dev);
  in_dev_hold(in_dev);
--
1.5.3.rc5
```

Subject: [PATCH 3/6] [NETNS]: Process interface address manipulation routines in the namespace.

Posted by den on Thu, 31 Jan 2008 12:00:47 GMT

View Forum Message <> Reply to Message

The namespace is available when required except rtm_to_ifaddr. Add
namespace argument to it.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 net/ipv4/devinet.c |   14 ++++++++------
 1 files changed, 8 insertions(+), 6 deletions(-)

diff --git a/net/ipv4/devinet.c b/net/ipv4/devinet.c
index e55c85e..6a6e92e 100644
```

```
--- a/net/ipv4/devinet.c
+++ b/net/ipv4/devinet.c
@@ -485,7 +485,7 @@ errout:
  return err;
 }

-static struct in_ifaddr *rtm_to_ifaddr(struct nlmsghdr *nlh)
+static struct in_ifaddr *rtm_to_ifaddr(struct net *net, struct nlmsghdr *nlh)
 {
  struct nlattr *tb[IFA_MAX+1];
  struct in_ifaddr *ifa;
@@ -503,7 +503,7 @@ static struct in_ifaddr *rtm_to_ifaddr(struct nlmsghdr *nlh)
  if (ifm->ifa_prefixlen > 32 || tb[IFA_LOCAL] == NULL)
   goto errout;

- dev = __dev_get_by_index(&init_net, ifm->ifa_index);
+ dev = __dev_get_by_index(net, ifm->ifa_index);
  err = -ENODEV;
  if (dev == NULL)
   goto errout;
@@ -571,7 +571,7 @@ static int inet_rtm_newaddr(struct sk_buff *skb, struct nlmsghdr *nlh, void
*arg
  if (net != &init_net)
   return -EINVAL;

- ifa = rtm_to_ifaddr(nlh);
+ ifa = rtm_to_ifaddr(net, nlh);
  if (IS_ERR(ifa))
   return PTR_ERR(ifa);

@@ -1189,7 +1189,7 @@ static int inet_dump_ifaddr(struct sk_buff *skb, struct netlink_callback
*cb)

  s_ip_idx = ip_idx = cb->args[1];
  idx = 0;
- for_each_netdev(&init_net, dev) {
+ for_each_netdev(net, dev) {
   if (idx < s_idx)
    goto cont;
   if (idx > s_idx)
@@ -1223,7 +1223,9 @@ static void rtmsg_ifa(int event, struct in_ifaddr* ifa, struct nlmsghdr
*nlh,
  struct sk_buff *skb;
  u32 seq = nlh ? nlh->nlmsg_seq : 0;
  int err = -ENOBUFS;
+ struct net *net;

+ net = ifa->ifa_dev->dev->nd_net;
```

```
  skb = nlmsg_new(inet_nlmsg_size(), GFP_KERNEL);
  if (skb == NULL)
   goto errout;
@@ -1235,10 +1237,10 @@ static void rtmsg_ifa(int event, struct in_ifaddr* ifa, struct nlmsghdr
*nlh,
   kfree_skb(skb);
   goto errout;
  }
- err = rtnl_notify(skb, &init_net, pid, RTNLGRP_IPV4_IFADDR, nlh, GFP_KERNEL);
+ err = rtnl_notify(skb, net, pid, RTNLGRP_IPV4_IFADDR, nlh, GFP_KERNEL);
 errout:
  if (err < 0)
-  rtnl_set_sk_err(&init_net, RTNLGRP_IPV4_IFADDR, err);
+  rtnl_set_sk_err(net, RTNLGRP_IPV4_IFADDR, err);
 }

 #ifdef CONFIG_SYSCTL
--
1.5.3.rc5
```

Subject: [PATCH 4/6] [IPV4]: fib_sync_down rework.
Posted by den on Thu, 31 Jan 2008 12:00:48 GMT
View Forum Message <> Reply to Message

fib_sync_down can be called with an address and with a device. In reality
it is called either with address OR with a device. The codepath inside is
completely different, so lets separate it into two calls for these two
cases.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 include/net/ip_fib.h     |    3 +-
 net/ipv4/fib_frontend.c  |    4 +-
 net/ipv4/fib_semantics.c |  104 ++++++++++++++++++++++++++----------------
 3 files changed, 57 insertions(+), 54 deletions(-)

diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h
index 9daa60b..1b2f008 100644
--- a/include/net/ip_fib.h
+++ b/include/net/ip_fib.h
@@ -218,7 +218,8 @@ extern void fib_select_default(struct net *net, const struct flowi *flp,

 /* Exported by fib_semantics.c */
 extern int ip_fib_check_default(__be32 gw, struct net_device *dev);
-extern int fib_sync_down(__be32 local, struct net_device *dev, int force);
+extern int fib_sync_down_dev(struct net_device *dev, int force);
+extern int fib_sync_down_addr(__be32 local);
```

```
 extern int fib_sync_up(struct net_device *dev);
 extern __be32  __fib_res_prefsrc(struct fib_result *res);
 extern void fib_select_multipath(const struct flowi *flp, struct fib_result *res);
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d0507f4..d69ffa2 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -808,7 +808,7 @@ static void fib_del_ifaddr(struct in_ifaddr *ifa)
       First of all, we scan fib_info list searching
       for stray nexthop entries, then ignite fib_flush.
    */
-   if (fib_sync_down(ifa->ifa_local, NULL, 0))
+   if (fib_sync_down_addr(ifa->ifa_local))
     fib_flush(dev->nd_net);
  }
 }
@@ -898,7 +898,7 @@ static void nl_fib_lookup_exit(struct net *net)

 static void fib_disable_ip(struct net_device *dev, int force)
 {
- if (fib_sync_down(0, dev, force))
+ if (fib_sync_down_dev(dev, force))
   fib_flush(dev->nd_net);
  rt_cache_flush(0);
  arp_ifdown(dev);
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index c791286..5beff2e 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -1031,70 +1031,72 @@ nla_put_failure:
      referring to it.
    - device went down -> we must shutdown all nexthops going via it.
  */
-
-int fib_sync_down(__be32 local, struct net_device *dev, int force)
+int fib_sync_down_addr(__be32 local)
 {
  int ret = 0;
- int scope = RT_SCOPE_NOWHERE;
-
- if (force)
-  scope = -1;
+ unsigned int hash = fib_laddr_hashfn(local);
+ struct hlist_head *head = &fib_info_laddrhash[hash];
+ struct hlist_node *node;
+ struct fib_info *fi;

- if (local && fib_info_laddrhash) {
```

```
-  unsigned int hash = fib_laddr_hashfn(local);
-  struct hlist_head *head = &fib_info_laddrhash[hash];
-  struct hlist_node *node;
-  struct fib_info *fi;
+ if (fib_info_laddrhash == NULL || local == 0)
+  return 0;

-  hlist_for_each_entry(fi, node, head, fib_lhash) {
-   if (fi->fib_prefsrc == local) {
-    fi->fib_flags |= RTNH_F_DEAD;
-    ret++;
-   }
+ hlist_for_each_entry(fi, node, head, fib_lhash) {
+  if (fi->fib_prefsrc == local) {
+   fi->fib_flags |= RTNH_F_DEAD;
+   ret++;
   }
  }
+ return ret;
+}

- if (dev) {
-  struct fib_info *prev_fi = NULL;
-  unsigned int hash = fib_devindex_hashfn(dev->ifindex);
-  struct hlist_head *head = &fib_info_devhash[hash];
-  struct hlist_node *node;
-  struct fib_nh *nh;
+int fib_sync_down_dev(struct net_device *dev, int force)
+{
+ int ret = 0;
+ int scope = RT_SCOPE_NOWHERE;
+ struct fib_info *prev_fi = NULL;
+ unsigned int hash = fib_devindex_hashfn(dev->ifindex);
+ struct hlist_head *head = &fib_info_devhash[hash];
+ struct hlist_node *node;
+ struct fib_nh *nh;

-  hlist_for_each_entry(nh, node, head, nh_hash) {
-   struct fib_info *fi = nh->nh_parent;
-   int dead;
+ if (force)
+  scope = -1;

-   BUG_ON(!fi->fib_nhs);
-   if (nh->nh_dev != dev || fi == prev_fi)
-    continue;
-   prev_fi = fi;
-   dead = 0;
```

```
-   change_nexthops(fi) {
-    if (nh->nh_flags&RTNH_F_DEAD)
-     dead++;
-    else if (nh->nh_dev == dev &&
-      nh->nh_scope != scope) {
-     nh->nh_flags |= RTNH_F_DEAD;
+ hlist_for_each_entry(nh, node, head, nh_hash) {
+  struct fib_info *fi = nh->nh_parent;
+  int dead;
+
+  BUG_ON(!fi->fib_nhs);
+  if (nh->nh_dev != dev || fi == prev_fi)
+   continue;
+  prev_fi = fi;
+  dead = 0;
+  change_nexthops(fi) {
+   if (nh->nh_flags&RTNH_F_DEAD)
+    dead++;
+   else if (nh->nh_dev == dev &&
+     nh->nh_scope != scope) {
+    nh->nh_flags |= RTNH_F_DEAD;
 #ifdef CONFIG_IP_ROUTE_MULTIPATH
-     spin_lock_bh(&fib_multipath_lock);
-     fi->fib_power -= nh->nh_power;
-     nh->nh_power = 0;
-     spin_unlock_bh(&fib_multipath_lock);
+    spin_lock_bh(&fib_multipath_lock);
+    fi->fib_power -= nh->nh_power;
+    nh->nh_power = 0;
+    spin_unlock_bh(&fib_multipath_lock);
 #endif
-     dead++;
-    }
+    dead++;
+   }
 #ifdef CONFIG_IP_ROUTE_MULTIPATH
-    if (force > 1 && nh->nh_dev == dev) {
-     dead = fi->fib_nhs;
-     break;
-    }
-#endif
-   } endfor_nexthops(fi)
-   if (dead == fi->fib_nhs) {
-    fi->fib_flags |= RTNH_F_DEAD;
-    ret++;
+   if (force > 1 && nh->nh_dev == dev) {
+    dead = fi->fib_nhs;
+    break;
```

```
   }
+#endif
+  } endfor_nexthops(fi)
+  if (dead == fi->fib_nhs) {
+   fi->fib_flags |= RTNH_F_DEAD;
+   ret++;
  }
 }

--
1.5.3.rc5
```

Subject: [PATCH 5/6] [NETNS]: Add a namespace mark to fib_info.
Posted by den on Thu, 31 Jan 2008 12:00:49 GMT
View Forum Message <> Reply to Message

This is required to make fib_info lookups namespace aware. In the other case
initial namespace devices are marked as dead in the local routing table
during other namespace stop.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 include/net/ip_fib.h     |    1 +
 net/ipv4/fib_semantics.c |    8 ++++----
 2 files changed, 5 insertions(+), 4 deletions(-)

diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h
index 1b2f008..cb0df37 100644
--- a/include/net/ip_fib.h
+++ b/include/net/ip_fib.h
@@ -69,6 +69,7 @@ struct fib_nh {
 struct fib_info {
   struct hlist_node fib_hash;
   struct hlist_node fib_lhash;
+ struct net  *fib_net;
   int   fib_treeref;
   atomic_t  fib_clntref;
   int   fib_dead;
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 5beff2e..97cc494 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -687,6 +687,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
   struct fib_info *fi = NULL;
   struct fib_info *ofi;
   int nhs = 1;
+ struct net *net = cfg->fc_nlinfo.nl_net;
```

```
  /* Fast check to catch the most weird cases */
  if (fib_props[cfg->fc_type].scope > cfg->fc_scope)
@@ -727,6 +728,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    goto failure;
  fib_info_cnt++;

+ fi->fib_net = net;
  fi->fib_protocol = cfg->fc_protocol;
  fi->fib_flags = cfg->fc_flags;
  fi->fib_priority = cfg->fc_priority;
@@ -798,8 +800,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
    if (nhs != 1 || nh->nh_gw)
     goto err_inval;
    nh->nh_scope = RT_SCOPE_NOWHERE;
-   nh->nh_dev = dev_get_by_index(cfg->fc_nlinfo.nl_net,
-           fi->fib_nh->nh_oif);
+   nh->nh_dev = dev_get_by_index(net, fi->fib_nh->nh_oif);
    err = -ENODEV;
    if (nh->nh_dev == NULL)
     goto failure;
@@ -813,8 +814,7 @@ struct fib_info *fib_create_info(struct fib_config *cfg)
  if (fi->fib_prefsrc) {
   if (cfg->fc_type != RTN_LOCAL || !cfg->fc_dst ||
       fi->fib_prefsrc != cfg->fc_dst)
-   if (inet_addr_type(cfg->fc_nlinfo.nl_net,
-        fi->fib_prefsrc) != RTN_LOCAL)
+   if (inet_addr_type(net, fi->fib_prefsrc) != RTN_LOCAL)
     goto err_inval;
  }

--
1.5.3.rc5
```

Subject: [PATCH 6/6] [NETNS]: Lookup in FIB semantic hashes taking into account the namespace.
Posted by den on Thu, 31 Jan 2008 12:00:50 GMT
View Forum Message <> Reply to Message

The namespace is not available in the fib_sync_down_addr, add it
as a parameter.

Looking up a device by the pointer to it is OK. Looking up using a result
from fib_trie/fib_hash table lookup is also safe. No need to fix that at all.
So, just fix lookup by address and insertion to the hash table path.

Signed-off-by: Denis V. Lunev <den@openvz.org>

```
---
 include/net/ip_fib.h     |    2 +-
 net/ipv4/fib_frontend.c  |    2 +-
 net/ipv4/fib_semantics.c |    6 +++++-
 3 files changed, 7 insertions(+), 3 deletions(-)

diff --git a/include/net/ip_fib.h b/include/net/ip_fib.h
index cb0df37..90d1175 100644
--- a/include/net/ip_fib.h
+++ b/include/net/ip_fib.h
@@ -220,7 +220,7 @@ extern void fib_select_default(struct net *net, const struct flowi *flp,
 /* Exported by fib_semantics.c */
 extern int ip_fib_check_default(__be32 gw, struct net_device *dev);
 extern int fib_sync_down_dev(struct net_device *dev, int force);
-extern int fib_sync_down_addr(__be32 local);
+extern int fib_sync_down_addr(struct net *net, __be32 local);
 extern int fib_sync_up(struct net_device *dev);
 extern __be32  __fib_res_prefsrc(struct fib_result *res);
 extern void fib_select_multipath(const struct flowi *flp, struct fib_result *res);
diff --git a/net/ipv4/fib_frontend.c b/net/ipv4/fib_frontend.c
index d69ffa2..86ff271 100644
--- a/net/ipv4/fib_frontend.c
+++ b/net/ipv4/fib_frontend.c
@@ -808,7 +808,7 @@ static void fib_del_ifaddr(struct in_ifaddr *ifa)
       First of all, we scan fib_info list searching
       for stray nexthop entries, then ignite fib_flush.
    */
-   if (fib_sync_down_addr(ifa->ifa_local))
+   if (fib_sync_down_addr(dev->nd_net, ifa->ifa_local))
     fib_flush(dev->nd_net);
  }
 }
diff --git a/net/ipv4/fib_semantics.c b/net/ipv4/fib_semantics.c
index 97cc494..a13c847 100644
--- a/net/ipv4/fib_semantics.c
+++ b/net/ipv4/fib_semantics.c
@@ -229,6 +229,8 @@ static struct fib_info *fib_find_info(const struct fib_info *nfi)
  head = &fib_info_hash[hash];

  hlist_for_each_entry(fi, node, head, fib_hash) {
+  if (fi->fib_net != nfi->fib_net)
+   continue;
   if (fi->fib_nhs != nfi->fib_nhs)
    continue;
   if (nfi->fib_protocol == fi->fib_protocol &&
@@ -1031,7 +1033,7 @@ nla_put_failure:
      referring to it.
    - device went down -> we must shutdown all nexthops going via it.
```

```
 */
-int fib_sync_down_addr(__be32 local)
+int fib_sync_down_addr(struct net *net, __be32 local)
 {
  int ret = 0;
  unsigned int hash = fib_laddr_hashfn(local);
@@ -1043,6 +1045,8 @@ int fib_sync_down_addr(__be32 local)
   return 0;

  hlist_for_each_entry(fi, node, head, fib_lhash) {
+  if (fi->fib_net != net)
+   continue;
   if (fi->fib_prefsrc == local) {
    fi->fib_flags |= RTNH_F_DEAD;
    ret++;
--
1.5.3.rc5
```

Subject: Re: [PATCH 1/6] [IPV4]: Fix memory leak on error path during FIB initialization.

Posted by davem on Thu, 31 Jan 2008 13:14:44 GMT

View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@openvz.org>
Date: Thu, 31 Jan 2008 15:00:45 +0300

> commit c8050bf6d84785a7edd2e81591e8f833231477e8
> Author: Denis V. Lunev <den@openvz.org>
> Date: Thu Jan 10 03:28:24 2008 -0800

I am fixing it up for you this time, but please do not
reference the commit this way.

Say something like:

  blah blah blah in commit $(SHA1_HASH) ("commit head line").

The author and date give no real useful information in
this context, the important part is giving the reader
enough information to find the commit should they wish
to gain more information.

If they have the commit hash they can usually find the
commit, but if that fails they can search the commit
messages for the head line text string.

I feel like I've had to explain this 10 times in the past week...

:-/

Subject: Re: [PATCH 0/6] preparations to enable netdevice notifiers inside a namespace (resend)
Posted by Daniel Lezcano on Thu, 31 Jan 2008 14:58:36 GMT
View Forum Message <> Reply to Message

Denis V. Lunev wrote:
> Here are some preparations and cleanups to enable network device/inet
> address notifiers inside a namespace.
>
> This set of patches has been originally sent last Friday. One cleanup
> patch from the original series is dropped as wrong, thanks to Daniel
> Lezcano.

Can you explain please.

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [PATCH 0/6] preparations to enable netdevice notifiers inside a namespace (resend)
Posted by Benjamin Thery on Thu, 31 Jan 2008 16:47:22 GMT
View Forum Message <> Reply to Message

On Jan 31, 2008 3:58 PM, Daniel Lezcano <daniel.lezcano@free.fr> wrote:

> Denis V. Lunev wrote:
> > Here are some preparations and cleanups to enable network device/inet
> > address notifiers inside a namespace.
> >
> > This set of patches has been originally sent last Friday. One cleanup
> > patch from the original series is dropped as wrong, thanks to Daniel
> > Lezcano.
>
> Can you explain please.

I think Denis refers to the patch called "3/7 Prohibit assignment of
0.0.0.0as interface address." ,
he dropped because it was inappropriate, no?

-- Benjamin

--
> To unsubscribe from this list: send the line "unsubscribe netdev" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
>

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 0/6] preparations to enable netdevice notifiers inside a namespace (resend)
Posted by Daniel Lezcano on Thu, 31 Jan 2008 17:46:46 GMT
View Forum Message <> Reply to Message

Benjamin Thery wrote:
> On Jan 31, 2008 3:58 PM, Daniel Lezcano <daniel.lezcano@free.fr> wrote:
>
>> Denis V. Lunev wrote:
>>> Here are some preparations and cleanups to enable network device/inet
>>> address notifiers inside a namespace.
>>>
>>> This set of patches has been originally sent last Friday. One cleanup
>>> patch from the original series is dropped as wrong, thanks to Daniel
>>> Lezcano.
>> Can you explain please.
>
>
> I think Denis refers to the patch called "3/7 Prohibit assignment of
> 0.0.0.0as interface address." ,
> he dropped because it was inappropriate, no?

Yes, you are right, Denis explained me in a private email. I think I
really need to sleep a little more :)

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [PATCH 0/6] preparations to enable netdevice notifiers inside a namespace (resend)
Posted by davem on Fri, 01 Feb 2008 02:50:14 GMT
View Forum Message <> Reply to Message

From: "Denis V. Lunev" <den@sw.ru>

Date: Thu, 31 Jan 2008 14:59:49 +0300

> Here are some preparations and cleanups to enable network device/inet
> address notifiers inside a namespace.
>
> This set of patches has been originally sent last Friday. One cleanup
> patch from the original series is dropped as wrong, thanks to Daniel
> Lezcano.

All applied, thanks.
