

---

Subject: [PATCH 1/4] add init logger support to vzctl  
Posted by [dietmar](#) on Tue, 20 Sep 2011 05:17:15 GMT

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

---

From: Dietmar Maurer <[dietmar@proxmox.com](mailto:dietmar@proxmox.com)>

Signed-off-by: Dietmar Maurer <[dietmar@proxmox.com](mailto:dietmar@proxmox.com)>

---

```
include/res.h      |  1 +
include/vzctl_param.h |  1 +
src/lib/config.c   | 10 ++++++
src/lib/env.c       | 80 ++++++++++++++++++++++++++++++++
4 files changed, 86 insertions(+), 6 deletions(-)
```

diff --git a/include/res.h b/include/res.h

index 2ad3d5b..e70be48 100644

--- a/include/res.h

+++ b/include/res.h

@@ -57,6 +57,7 @@ typedef struct {

int onboot;

unsigned long \*bootorder;

int wait;

+ int initlog;

} misc\_param;

struct mod\_action;

diff --git a/include/vzctl\_param.h b/include/vzctl\_param.h

index ff520da..9245b6a 100644

--- a/include/vzctl\_param.h

+++ b/include/vzctl\_param.h

@@ -136,6 +136,7 @@

#define PARAM\_BOOTORDER 364

#define PARAM\_PCI\_ADD 365

#define PARAM\_PCI\_DEL 366

+#define PARAM\_INITLOG 367

#define PARAM\_LINE "e:p:f:t:i:l:k:a:b:n:x:h"

#endif

diff --git a/src/lib/config.c b/src/lib/config.c

index baef108..bcbfb5 100644

--- a/src/lib/config.c

+++ b/src/lib/config.c

@@ -62,6 +62,7 @@ static vps\_config config[] = {

{"LOG\_LEVEL", NULL, PARAM\_LOGLEVEL},

{"LOGFILE", NULL, PARAM\_LOGFILE},

{"VERBOSE", NULL, PARAM\_VERBOSE},

{"INITLOG", NULL, PARAM\_INITLOG},

```

{"IPTABLES", NULL, PARAM_IPTABLES},
/* UB */
@@ -121,6 +122,8 @@ static vps_config config[] = {
 {"CPULIMIT", NULL, PARAM_CPULIMIT},
 {"CPUS", NULL, PARAM_VCPUS},
 {"CPUMASK", NULL, PARAM_CPUMASK},
+/* misc param */
+{"INITLOG", NULL, PARAM_INITLOG},
 /* create param */
 {"ONBOOT", NULL, PARAM_ONBOOT},
 {"CONFIGFILE", NULL, PARAM_CONFIG},
@@ -1273,6 +1276,9 @@ static int store_misc(vps_param *old_p, vps_param *vps_p,
vps_config *conf,
ret = 0;
switch (conf->id) {
+ case PARAM_INITLOG:
+ ret = conf_store_yesno(conf_h, conf->name, misc->initlog);
+ break;
case PARAM_ONBOOT:
ret = conf_store_yesno(conf_h, conf->name, misc->onboot);
break;
@@ -1887,6 +1893,9 @@ static int parse(envid_t veid, vps_param *vps_p, char *val, int id)
case PARAM_LOGGING:
ret = conf_parse_yesno(&vps_p->log.enable, val);
break;
+ case PARAM_INITLOG:
+ ret = conf_parse_yesno(&vps_p->res.misc.initlog, val);
+ break;
case PARAM_LOGLEVEL:
if (parse_int(val, &int_id))
return ERR_INVAL;
@@ -2710,6 +2719,7 @@ static void merge_misc(misc_param *dst, misc_param *src)
MERGE_INT(onboot)
MERGE_P(bootorder)
MERGE_INT(wait)
+ MERGE_INT(initlog)
}

static void merge_dq(dq_param *dst, dq_param *src)
diff --git a/src/lib/env.c b/src/lib/env.c
index 5e53657..135286f 100644
--- a/src/lib/env.c
+++ b/src/lib/env.c
@@ -23,6 +23,7 @@
#include <signal.h>
#include <fcntl.h>
```

```

#include <sys/types.h>
+#include <sys/stat.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <string.h>
@@ -318,7 +319,7 @@ static int _env_create(vps_handler *h, envid_t veid, int wait_p, int err_p,
int fd, ret;
vps_res *res;
char *argv[] = {"init", "-z", "", NULL};
- char *envp[] = {"HOME=/", "TERM=linux", NULL};
+ char *envp[] = {"HOME=/", "TERM=linux", NULL, NULL};

res = (vps_res *) data;
memset(&create_param, 0, sizeof(create_param));
@@ -425,12 +426,65 @@ try:
*/
if (read(wait_p, &ret, sizeof(ret)) == 0)
return 0;
- if ((fd = open("/dev/null", O_RDWR)) != -1) {
- dup2(fd, 0);
- dup2(fd, 1);
- dup2(fd, 2);
- }
+
logger(10, 0, "Starting init");
+
+ if (((fd = open("/dev/null", O_RDWR)) != -1) && (fd == STDIN_FILENO)) {
+
+ dup2(fd, 1); // STDOUT = /dev/null
+ dup2(fd, 2); // STDERR = /dev/null
+
+ if (res->misc.initlog == 1) {
+     int pid;
+
+     if (!((mkfifo ("/var/log/init.fifo", 0600) == 0) || (errno == EEXIST))) {
+         logger(-1, errno, "unable to create init fifo");
+         return 0;
+     }
+
+     // create new /dev/console ==> /var/log/init.fifo
+     unlink ("/dev/console");
+     symlink ("/var/log/init.fifo", "/dev/console");
+
+     pid = fork();
+
+     if (pid == -1) {
+         logger(-1, errno, "unable to fork init-logger");
+     }
+ }
}

```

```

+ }
+
+ if (!pid) {
+
+     close(wait_p); close(err_p);
+
+ /* Note: open fifo RDWR to avoid EOF */
+ close(STDIN_FILENO);
+ open("/var/log/init fifo", O_RDWR);
+
+ close(STDOUT_FILENO);
+ open("/var/log/init.log", O_CREAT|O_WRONLY|O_TRUNC);
+
+ close(STDERR_FILENO);
+ dup2(STDOUT_FILENO, STDERR_FILENO);
+
+ execl("/sbin/init-logger", NULL);
+ exit (-1);
+ }
+
+ close(STDOUT_FILENO);
+ if (open("/var/log/init fifo", O_WRONLY) == -1) {
+     logger(-1, errno, "open init fifo failed");
+     return 0;
+ }
+
+ close(STDERR_FILENO);
+ dup2(STDOUT_FILENO, STDERR_FILENO);
+
+ // tell init to use /var/log/init fifo
+ envp[2] = "CONSOLE=/var/log/init fifo";
+ }
+
execve("/sbin/init", argv, envp);
execve("/etc/init", argv, envp);
execve("/bin/init", argv, envp);
@@ @ -446,6 +500,20 @@ static int vz_real_env_create(vps_handler *h, evid_t veid, vps_res
*res,
{
    int ret, pid;

+     char ildest[4096];
+
+     *ildest = 0;
+     strcat (ildest, res->fs.root);
+     strcat (ildest, "/sbin/init");
+

```

```
+ *ildest = 0;
+ strcat (ildest, res->fs.root);
+ strcat (ildest, "/sbin/init-logger");
+ if (cp_file (ildest, "/usr/lib/vzctl/scripts/init-logger") != 0) {
+     logger(-1, 0, "Unable to copy init-logger");
+     return VZ_RESOURCE_ERROR;
+ }
+
if ((ret = vz_chroot(res->fs.root)))
    return ret;
if ((ret = vz_setluid(veid)))
--
```

## 1.5.6.5

---