Quo Vadis bvol

Richard Hartmann
(rh@globalways.net)

Globalways AG

September 22, 2010

1. Life with Bacula
2. Host management
3. bvols on disk
4. Automagic configuration
5. Verify everything
6. Catalog backup

1 Life with Bacula

2 Host management

3 bvols on disk

4 Automagic configuration

5 Verify everything

6 Catalog backup

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
    - Stable
    - Scalable
    - Flexible
    - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
    - Stable
    - Scalable
    - Flexible
    - Fast
- We love it
- Yet, not everything is perfect
- Side notes
    - Still using Bacula 3
    - Each Director and its Storage are a self-contained unit

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes
  - Still using Bacula 3
  - Each Director and its Storage are a self-contained unit

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes
  - Still using Bacula 3
  - Each Director and its Storage are a self-contained unit

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes
  - Still using Bacula 3
  - Each Director and its Storage are a self-contained unit

Our experience

- We used to run a set of self-baked scripts...
- We migrated to Bacula because it's:
  - Stable
  - Scalable
  - Flexible
  - Fast
- We love it
- Yet, not everything is perfect
- Side notes
  - Still using Bacula 3
  - Each Director and its Storage are a self-contained unit

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
  - Situations in which no backup was made at all!
  - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
  - Situations in which no backup was made at all!
  - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
    - Situations in which no backup was made at all!
    - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
    - Situations in which no backup was made at all!
    - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
  - Situations in which no backup was made at all!
  - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

Issues that annoyed us

- Designed to work with tapes
- Manual configuration does not scale
- No review of backup jobs
  - Situations in which no backup was made at all!
  - Not merely an in-house issue; patches sent
- Catalog backups take too long to replay

1. Life with Bacula
2. **Host management**
3. bvols on disk
4. Automagic configuration
5. Verify everything
6. Catalog backup

Management/awareness is everything

- In-house host management system
- UUIDs for everything (sample UUID: 1f39f0a9-0172-4273-93fc-2e393fd5d5b4)
- UUIDs are for computers, not people; need for abstraction and automation
- All solutions need to tie in to our management framework
- Aggressive monitoring and verification

Management/awareness is everything

- In-house host management system
- UUIDs for everything (sample UUID: 1f39f0a9-0172-4273-93fc-2e393fd5d5b4)
- UUIDs are for computers, not people; need for abstraction and automation
- All solutions need to tie in to our management framework
- Aggressive monitoring and verification

Management/awareness is everything

- In-house host management system
- UUIDs for everything (sample UUID: 1f39f0a9-0172-4273-93fc-2e393fd5d5b4)
- UUIDs are for computers, not people; need for abstraction and automation
- All solutions need to tie in to our management framework
- Aggressive monitoring and verification

Management/awareness is everything

- In-house host management system
- UUIDs for everything (sample UUID: 1f39f0a9-0172-4273-93fc-2e393fd5d5b4)
- UUIDs are for computers, not people; need for abstraction and automation
- All solutions need to tie in to our management framework
- Aggressive monitoring and verification

Management/awareness is everything

- In-house host management system
- UUIDs for everything (sample UUID: 1f39f0a9-0172-4273-93fc-2e393fd5d5b4)
- UUIDs are for computers, not people; need for abstraction and automation
- All solutions need to tie in to our management framework
- Aggressive monitoring and verification

1 Life with Bacula

2 Host management

3 bvols on disk

4 Automagic configuration

5 Verify everything

6 Catalog backup

"One bvol is one tape"

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

”One bvol is one tape”

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

"One bvol is one tape"

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

"One bvol is one tape"

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

"One bvol is one tape"

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

"One bvol is one tape"

- Bacula has been designed with tapes in mind
  - Not necessarily a downside
  - Still, not suitable for us
- Recycling storage within bvols can spread backups across several bvols
- Depending on the layout, there is no clean separation between customers, hosts, file sets, and backup jobs
- No direct access from our management system

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure:
  client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure:
  client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure: client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

"One bvol per backup job"

- Solution: Create one bvol per backup job
- Flexible
- Clear directory structure:
  client-id/host-uuid/client-id_host-uuid_fileset_level_date
- Most granular form of backup storage
- Outside verification of:
  - existence
  - expected file sizes
  - quotas per customer, project, or host
- Simply delete bvols when they get stale

1 Life with Bacula

2 Host management

3 bvols on disk

4 Automagic configuration

5 Verify everything

6 Catalog backup

We are not Sisyphus; neither should you

- Manual configuration does not scale
  - Takes time
  - Error-prone
  - Hard to verify programmatically
  - We are using UUIDs, not names
- Database-driven config generation

We are not Sisyphus; neither should you

- Manual configuration does not scale
  - Takes time
  - Error-prone
  - Hard to verify programmatically
  - We are using UUIDs, not names
- Database-driven config generation

We are not Sisyphus; neither should you

- Manual configuration does not scale
  - Takes time
  - Error-prone
  - Hard to verify programmatically
  - We are using UUIDs, not names
- Database-driven config generation

We are not Sisyphus; neither should you

- Manual configuration does not scale
    - Takes time
    - Error-prone
    - Hard to verify programmatically
    - We are using UUIDs, not names
- Database-driven config generation

We are not Sisyphus; neither should you

- Manual configuration does not scale
  - Takes time
  - Error-prone
  - Hard to verify programmatically
  - We are using UUIDs, not names
- Database-driven config generation

We are not Sisyphus; neither should you

- Manual configuration does not scale
  - Takes time
  - Error-prone
  - Hard to verify programmatically
  - We are using UUIDs, not names
- Database-driven config generation

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package

- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML-RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

From zero to bacula.conf

- Create host in host manager
- Spawn VM from template
- Retrieve UUID from host, copy & paste into host manager
- Select schedule and fileset for host
- Install custom Bacula package
  - Sends authentication and host-UUID via XML RPC
  - Receives configuration
  - Finishes local installation
- The same happens on Storage and Director on demand and with verification
- No user interaction required to set up Bacula!

1 Life with Bacula

2 Host management

3 bvols on disk

4 Automagic configuration

5 Verify everything

6 Catalog backup

Backups are live-savers; treat them as such

- Custom MySQL plugin to prohibit remote execution of commands
  - Module failed to load but backup "OK"; patch sent upstream
  - Uncaught bug in our module, thus no backup
- Lesson learned: paranoia is good!

Backups are live-savers; treat them as such

- Custom MySQL plugin to prohibit remote execution of commands
  - Module failed to load but backup "OK"; patch sent upstream
  - Uncaught bug in our module, thus no backup
- Lesson learned: paranoia is good!

Backups are live-savers; treat them as such

- Custom MySQL plugin to prohibit remote execution of commands
  - Module failed to load but backup "OK"; patch sent upstream
  - Uncaught bug in our module, thus no backup
- Lesson learned: paranoia is good!

Backups are live-savers; treat them as such

- Custom MySQL plugin to prohibit remote execution of commands
  - Module failed to load but backup "OK"; patch sent upstream
  - Uncaught bug in our module, thus no backup
- Lesson learned: paranoia is good!

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

Our additional safe-guards

- We introduced additional checks if
  - expected bvols exist
    - monitoring reacts in five minutes or less when new hosts are configured
  - bvols' sizes fit predictions
  - jobs are stuck in state Created for too long
  - any job is not in state Running, Terminated normally, or Canceled
  - non-fatal errors occured
- All data is fed into our monitoring
- Every single event needs to be looked at, verified, commented and closed manually
- Important backups are stored in three separate, remote locations

1. Life with Bacula
2. Host management
3. bvols on disk
4. Automagic configuration
5. Verify everything
6. Catalog backup

Is this even a section?

- MySQL hot-copy
- Locks database!
- Takes 20% more space
- Backup faster by 30%
- Lightning fast restoration

Is this even a section?

- MySQL hot-copy
- Locks database!
- Takes 20% more space
- Backup faster by 30%
- Lightning fast restoration

Is this even a section?

- MySQL hot-copy
- Locks database!
- Takes 20% more space
- Backup faster by 30%
- Lightning fast restoration

Is this even a section?

- MySQL hot-copy
- Locks database!
- Takes 20% more space
- Backup faster by 30%
- Lightning fast restoration

Is this even a section?

- MySQL hot-copy
- Locks database!
- Takes 20% more space
- Backup faster by 30%
- Lightning fast restoration

The End

Thank you for your attention!

Richard Hartmann
Globalways AG
rh@globalways.net

RichiH @ irc://irc.freenode.net/bacula