

How-to Guide
SAP NetWeaver 2004s



How To Minimize Effects of Planned Downtime

December 2006

Applicable Releases:
SAP NetWeaver 2004
SAP NetWeaver 2004s

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Contents

Contents	1
Abstract.....	2
High Availability (HA).....	2
Planned Downtime	2
The Myth of the Nines	3
The Challenge of Continuous Availability	3
Minimizing Effects of Planned Downtime	4
General Idea	5
General Principles.....	5
Goals	8
Show Case: Minimizing Planned Downtime on Solaris 10.....	8
Virtualization	9
Why using Virtualization for SAP Systems?	9
Why not using Virtualization for SAP Systems?	9
P2V/ V2P	10
Choosing a virtualization technique/ level/ product.....	10
Scenario.....	11
Environment.....	11
Architecture.....	11
Effective Downtime.....	15
Step-by-Step approach.....	15
Automated Procedure.....	17
Quick Review	18
Further Considerations	18
Conclusion	19
Limitations.....	19
Outlook	19
Appendix	21
Terminology	21
Tables and Figures.....	21

Abstract

In the last year companies faced the growing challenge of centralizing and harmonizing their IT systems. As a consequence of this fact the requirements for the system availability increased dramatically: When using a system for 24 hours a day, 7 days a week and 52 weeks a year there are no timeslots for updates or tests. According to these requirements the costs of downtime are growing year by year [i].

Therefore new concepts need to be applied to increase system availability and to minimize the effects of downtime to a productive system this way.

This document gives an overview on how to minimize planned downtime using a shadow system approach.

High Availability (HA)

An unplanned downtime occurs if an application crashes unexpectedly. In many cases this is caused by a hardware fault [ii]. Other repetitive reasons are a crash of the host operating system or human failure for example.

The Institute of Electrical and Electronics Engineers (IEEE) defines High Availability as follows:

“High Availability (HA for short) refers to the availability of resources in a computer system, in the wake of component failures in the system.” [iii]

To achieve a scalable and flexible system landscape professional application servers are usually installed in a software cluster today. This means that there are various instances of the same service, available on different physical machines. These physical instances are usually reached via a load balancer which dispatches requests from the clients. Besides dispatching the workload this concept also provides a redundant infrastructure which enables resistance to the most hardware and software faults.

Despite this redundancy in many system landscapes there are still services which are unique within the cluster environment and vital for the cluster operation. If one of these services would crash the whole cluster might not work any more. These central, non-redundant services are referred to as potential Single Point of Failures (SPOF). High Availability solutions provide functionalities to make these SPOF's available continuously. In current versions of the most mainstream operating systems a high availability solution is integrated by default or can be integrated using 3rd party software.

Since SAP NetWeaver 2004 SAP supports High Availability setups out of the box [iv] by isolating potential SPOF's. Concepts for providing high availability of these potential SPOF's are provided by SAP's partners pending on the scenario and platform.

Planned Downtime

Beside unplanned downtime there are situations which require a restart or temporary downtime of the application itself. Some examples are updates, security fixes, patches, or tests. These situations are referred to as Planned Downtime.

Compared to the problem of unplanned downtime these situations are quite more complex to handle – for example, if an update triggers some fundamental architectural changes it has to be applied to all nested applications.

The Myth of the Nines

In today's Service Level Agreements (SLA) the system availability is usually measured in percentage uptime per year as shown in the table below.

Availability (%)	Downtime per year
99.9	8.76 hours
99.99	52.6 minutes
99.999	5.26 minutes
99.9999	32 seconds
99.99999	3.5 seconds

Table 1: System Availability Matrix

These SLA's are usually created by hosting providers and computing centers for their customers. In general, they are defining the term "High Availability" as an availability starting at 99.99% (53 minutes per year). Currently, an availability of 99,999% is the best availability claimed for commercial systems.

Despite these figures may look pretty impressive to customers the statement is only valid for unplanned downtime. It does not include any planned downtime estimations but assumes that an application does not change after it is once installed.

Offline Backups, Upgrades, Updates, Security Fixes to Applications and operating systems and so on are not included in these calculations.

The Challenge of Continuous Availability

Apart from Unplanned Downtime, the reasons for Planned Downtime are usually not external faults like hardware failures. Instead planned downtime is describing a downtime caused by a required offline maintenance operation of an application and the hosting environment, such as a DB or an OS, itself.

Example: Security Fixes need to be installed and the whole application server might have to be restarted afterwards.

Figure 1 provides a short overview about common planned downtime estimations for SAP system landscapes.

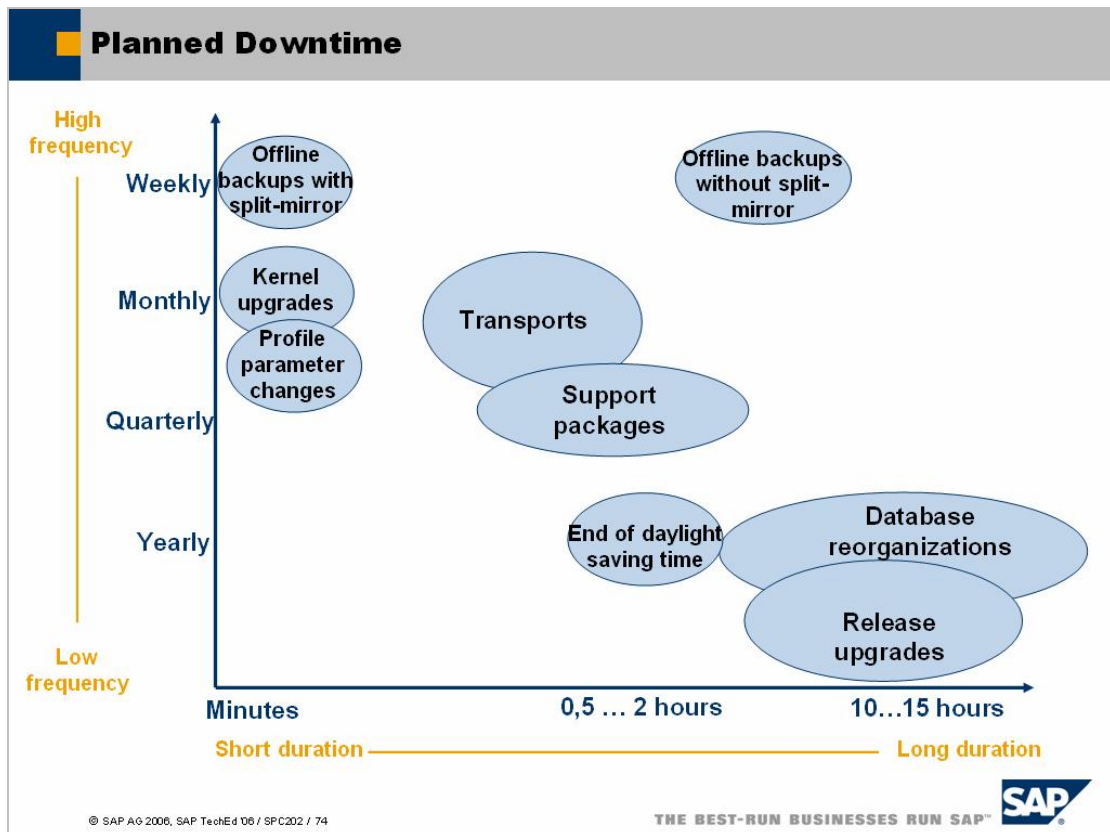


Figure 1: Planned Downtime Estimations

Historically, the term “High Availability System” was introduced by hardware solutions clearly tackling unplanned downtime caused by hardware outages. Therefore, systems rated as HA are not meant to be protected against downtime caused by planned activities. So obviously, excluding planned downtime activities in HA is not a limitation of HA but a historical misconception. The purpose of course is to provide reliability and uninterrupted business in general.

For example, an unplanned downtime of a SAP system during the working hours would probably have more impact to the operations of a company than a planned downtime on Sunday.

However, customers today are not only looking for unplanned downtime protection (High Availability Systems). They are also asking for systems that are “continuous available” to fit the needs required by a centralized IT infrastructure. While Continuous Availability can hardly be achieved at all with current software and hardware technologies, it is considered as a long term goal for most of current IT developments.

Minimizing Effects of Planned Downtime

High Availability setups of critical business applications have become a standard in the last years. Although unplanned downtime need to be handled with care the impacts of these downtime are mostly manageable today by using robust HA cluster software/hardware and setting up the system appropriately.

Other than supporting HA solutions, providing a continuous software environment requires dramatic architectural changes in the current software design. Also providing solutions which are applicable for already existing system landscapes is obligatory.

General Idea

The concept described here is based on a kind of shadow system: An existing SAP system is copied in online mode using an appropriate cloning mechanism. Afterwards the cloned system is started independently from the productive system. Now the operation which would normally cause a planned downtime (e.g. update) can be applied as shown in the picture below.

Please note that during step 2 the productive system should run in a read-only mode because all changes applied to the system during this point of time would be transient and lost after Step 3 was performed. Of course, this fact is a show stopper for some scenarios. However, there are other cases where these limitations would be acceptable (a typical Enterprise Portal setup for example).

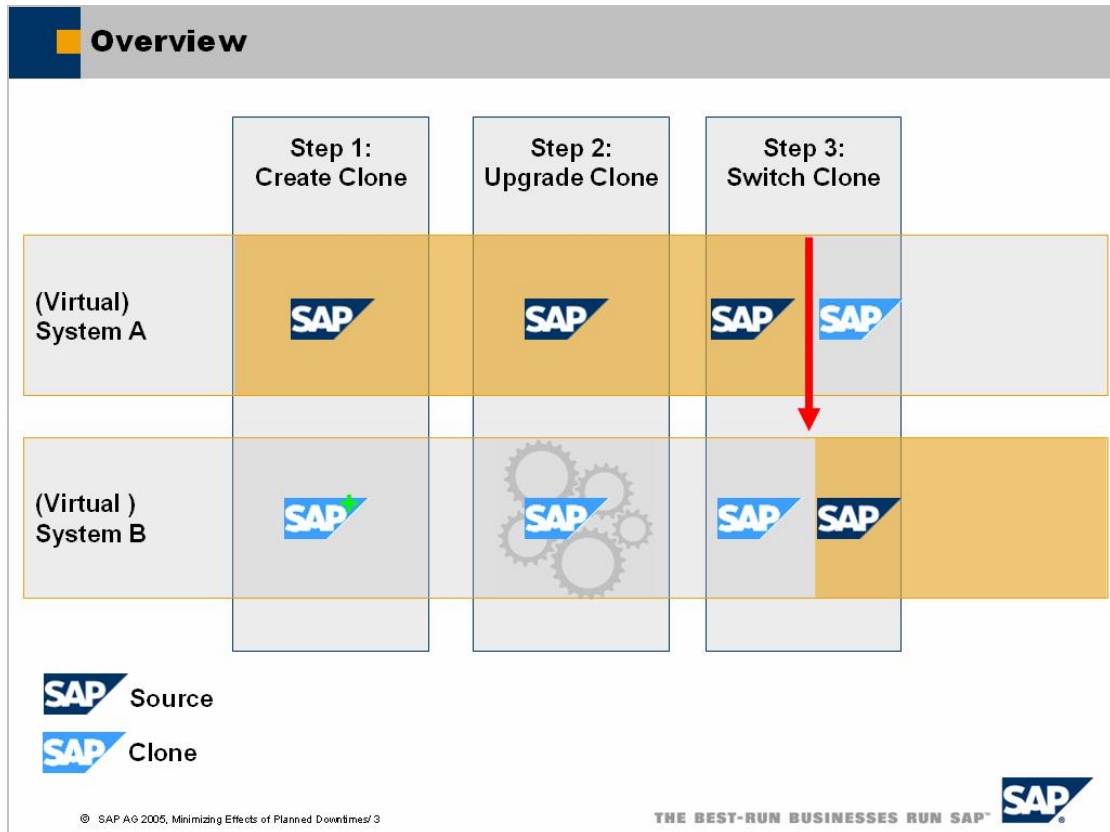


Figure 2: Shadow System Approach

After the operation on the shadow system finished successfully the systems are switched and the roles are changed. This means that System B becomes the new productive system afterwards. This way no effective downtime of the productive system is required.

General Principles

Today there are various ways of minimizing the effects that are caused by planned system downtime. One way is the shadow system approach described here. Figure 3 shows some general building blocks which can be applied for creating a system landscape using the shadow system approach described above. Applying these principles should be possible in the most scenarios using appropriate hardware and software techniques.

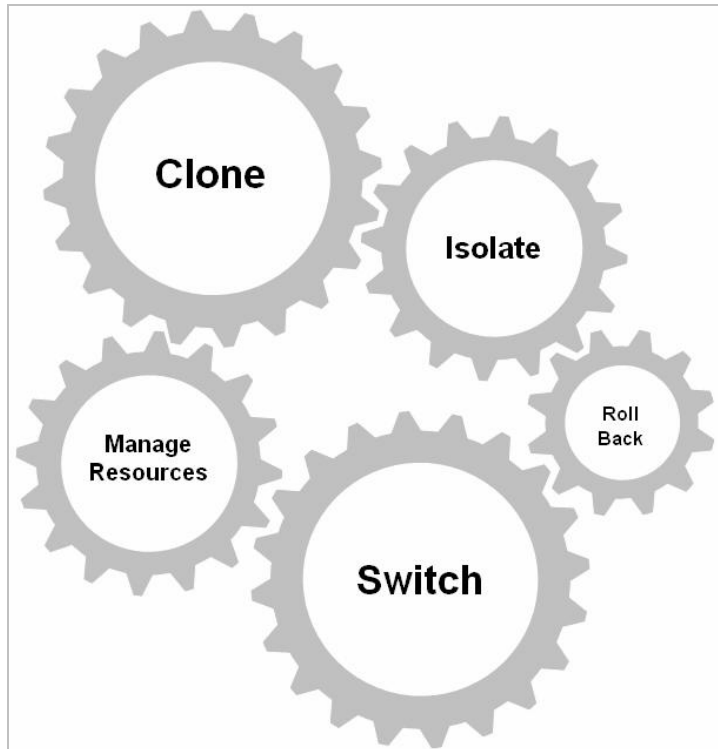


Figure 3: Principles for Minimizing Planned Downtime



Clone Capability

For creating a shadow system, a copy of the productive system is required. To create such a copy in online mode a cloning technique could be used.

The cloning technique should fit the following requirements

- Consistent (online) cloning mechanism
- Writable access to the source file system and the clone file system at the same point of time
- Data can be changed in both systems independently without affecting each other (see isolation also)

Cloning capabilities may be implemented via hardware or software solution. If the shadow system is determined to run on another physical host the clone needs to be designed in a transportable manner, too. If the clone is not transportable, SAP's System Copy mechanism might help for transporting the clone.



Switch Capability

A switch is considered to be a mechanism which changes the role of the productive system and the shadow system: The productive system takes over the role of the shadow system and vice versa after an update was applied successfully to the shadow system.

A switch should be performed fast, consistent, reliable and deterministic. Also the switch needs to be performed in a transactional manner. This means that when switching the systems all clients should be switched to the new system at the same point of time to prevent mixed system landscapes.



Isolate Capability

When creating a shadow system no extensive adoption of the system landscape should be necessary. For example, renaming the hostname on which the SAP system is pending on should not be required.

Also it is extremely important to isolate the productive system from the shadow system. Some examples:

- Accessing other backend systems at the same point of time from the productive and the shadow system should be restricted (e.g. Read-Only)
- Network access must be isolated properly
- Data must be separated properly
- Resources like ports, services and so on needs to be isolated if the clone is determined to run on the same physical host
- Additional considerations are required. For instance: What is the best way to handle existing batch jobs, printer jobs and so on.

Changes applied to the shadow system must not affect the productive system.



Manage Resources Capability

In order to provide an uninterrupted and reliable business the shadow system must not affect the productive system's operations.

For example, if the systems are located on the same hardware it is necessary to ensure that the operation of the productive system is not affected by the shadow system. Therefore it should be possible to limit the workload including CPU, memory, I/O request, network load and so on.

Further information regarding these requirements can be found within the SAP System Copy guide, too.



Online Rollback Capability

In rare cases when upgrading a system unpredictable errors may occur. If the update of the shadow system fails it should be possible to get back to the old system without a downtime of the productive system.

Of course, this could be done by using a conventional backup, too. But creating the backup and rolling back the system would cause significant downtime depending on the amount of data. Also usually in a backup only the file system is included and can be rolled back. It does not include the OS level data like users, services and so on. This is usually reached by separating the productive and the shadow system as already described in the isolation section.

Figure 4 shows the building block sequence.

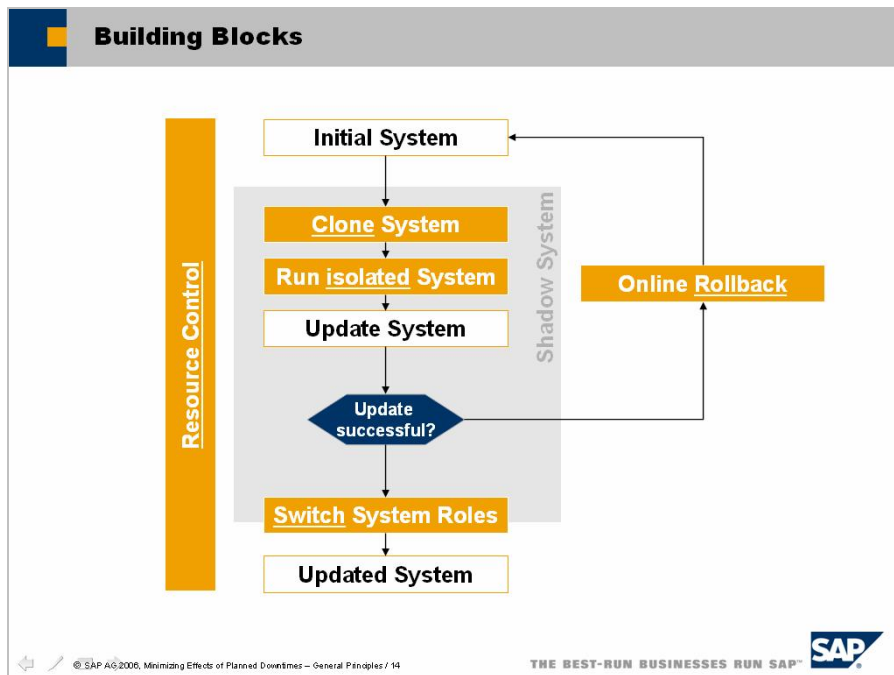


Figure 4: Sequence of Building Blocks

Combining and implementing these principles a shadow system can be implemented which may reduce the downtime of a productive system dramatically.

Goals

There are various reasons why a setup described in this scenario here can help to reduce the effects of downtime including:

Goal	Example
Access point to subsystems during planned downtime	An Enterprise Portal is used as a user interface for a mySAP ERP system.
Access point to information content during planned downtime	Reading information content stored within Knowledge Management.
Provides a rollback point after an update	If an update or system change results in a crash of the system, the read-only system can be used as a rollback point (no additional offline backup is required).
Creates a flexible test environment without downtime	Before applying new software to the productive system this software can be tested in the productive landscape without any downtime or risk.

Show Case: Minimizing Planned Downtime on Solaris 10

To give an idea how the principles described above can be applied in practice, this show case demonstrates how to use Sun Solaris 10 build in features to minimize effects of planned downtime. These features include the Zone virtualization concept introduced with Solaris 10.

Zones are similar to BSD Jails or Linux vServer and can be described as a kind of extended chroot environment integrating advanced security and resource management features. Zones allow creating a dedicated OS instance within a Solaris 10 host system which is isolated from the host operating system and also isolated from other zones.

Please note that the setup described in this show case is just show case for the shadow system approach: The general concepts can be applied to every platform. Due to the specific implementation depends on the project scenario, customers how are planning to implement this kind of setup should consult the associated hardware or software partner.

To understand how virtualization technology can be used for supporting the shadow system approach some aspects of virtualization are considered in the following.

Virtualization

Although virtualization has become a buzzword within the last years the concept is not new. The first virtual machine based on IBM's VM/CMS was created decades ago. Today a virtual machine like the Java VM is installed on the most of the computers in the world.

Independent of the specific implementation, virtualization products can be grouped in two different approaches which are Virtual Machines (using Hypervisors or Paravirtualization techniques like VMWare or XEN) and Virtual Servers (using OS partitioning like Linux VServer or Solaris Zones) roughly.

Additionally, a lot of architectures also provide hardware support for virtualization. Since AMD and Intel introducing native hardware support for virtualization within their current processors, virtualization on these platforms will become faster and easier also on the x86(_64) architecture.

Why using Virtualization for SAP Systems?

The benefits of installing applications such as SAP in virtualized environments depend on the business case. In general there are some general advantages which should be valid for the most scenarios including

- + Increasing system stability and security using isolation of different systems, e.g. separating a database server from the SAP application server using different zones and hardware pools
- + Lower costs for computing centers/ administration
- + Higher Flexibility (e.g. transportability)
- + New ways for Backups, Tests, Maintenance, Upgrades, Patches etc.

For more general advantages of virtualization take a look at [\[v\]](#).

Why not using Virtualization for SAP Systems?

Beside the advantages listed above there are some disadvantages when using virtualization techniques including:

- Performance loss caused by virtualization (up to ~40% depending on the product and the implementation of the solution)
- Additional training, configuration and administration efforts are required
- Virtualization software may cause unpredictable errors
- Host system is a potential single point of failure (SPOF) for all hosted systems
- Limited support by SAP [\[vi\]](#)

P2V/ V2P

Many system landscapes do not implement virtualization concepts at the moment. To use the benefits of virtualization not only in new setups but also in existing environments a way to migrate existing content of physical hosts into virtualized hosts is needed. This is often referred to as P2V (physical-to-virtual). Migrating virtualized hosts to physical machines is also possible using V2P (virtual-to-physical). These concepts could be interesting for transporting, cloning and upgrading systems as well.

Some vendors of virtualization products already offerings best practices [vii] and products [viii] for enabling P2V transport.

Choosing a virtualization technique/ level/ product

There is a wide variety of virtualization products available on the market using different virtualization levels and concepts. A list of different solutions and the used techniques can be found at [ix].

All these products may have their very special benefits and disadvantages. The optimal product depends on the requirements of the specific scenario. For example, virtual servers do usually have a good quality in latency and performance. On the other hand they are not as flexible as many other solutions using hypervisor techniques.

Also there are products that are well approved already on the market due to years of productive usage. Other products are not rated as ready for productive use in general. If you want to get a short general overview about current virtualization methods you may take a look at [x].

The picture below is based on x86(_64) architecture but should be valid for a lot of other hardware platforms as well.

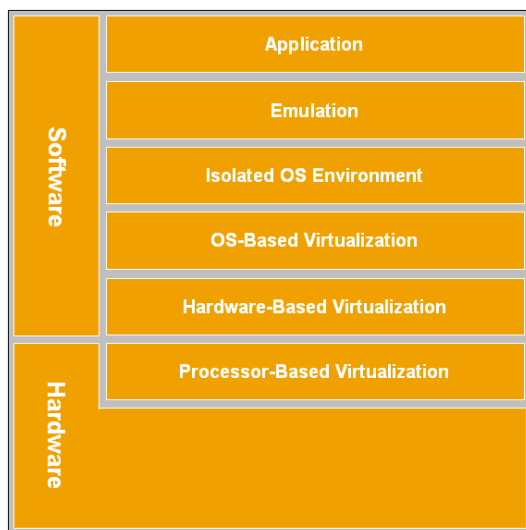


Figure 5: Virtualization Levels

Current virtualization technology may provide some of the principles just described out of the box. Anyway, of course virtualization is just one way and not the only one for the implementation.

As already mentioned above in this scenario Solaris Zones are used as virtualization technique for isolating the productive system and the shadow system.

Scenario

The scenario described in this document is using the shadow system approach: During the update of a productive NetWeaver system a read-only system is created which is still available for the clients. This way the impact of applying an update to a productive system is decreased. For example, an Enterprise Portal which is currently in read-only mode can still act as an access point to an associated ERP system.

The scenario consists of 3 key steps as already shown in Figure 2.

First, system B by cloning the productive system A is created. System B is used for applying the update later. During Step 1 and Step 2 the source system (A) is in a transient state which means that all changes applied to the system during this timeframe will be discarded after switching the roles of the systems. Therefore a read-only phase for the system is obligatory. Because not all applications are supporting a read-only mode as build-in functionality an organizational setup of this mode should be considered, too. For example, a content freeze during the read-only time could be announced before the setup starts.

As soon as the update was applied successfully to system B and the tests finished, the switch of the systems can be performed (Step 3): The roles of system A and system B are switched which means that B is the new source system and A is the “clone” afterwards. System B is representing the productive system now.

Please take a look into Appendix→Terminology for not getting confused with the names used to specify the systems.

Environment

The setup described here was implemented on a SAP NetWeaver 04s SR1 system running on Sun Solaris 10 installed on a Sun Fire-V490.

Sun Solaris 10 includes the following features which were used within this scenario:

- **Solaris Zones:** More information regarding Zones is available at [xi].
- **Solaris Resource Management:** The Resource Management feature enables the limitation and assignment of hardware resources such as CPU and memory by the usage of so called pools. These pools can be assigned to a zone directly [xii].
- **Solaris Zettabyte File System:** The Zettabyte File System (ZFS) integrates a snapshot facility which enables the creation of a clone based on a ZFS file system structure within a few seconds. This feature enables cloning a SAP system without any downtime. For further information about ZFS take a look at [xiii].

Within the “Quick Review” on page 18 of this document the use of the technology listed above are summarized.

To get a short overview about the pros and cons for installing a SAP system in Solaris Zones take a look at [xiv]. You will find some general hints for installing SAP in a zone environment there, too.

Architecture

Before describing the detailed steps a general understanding of the implemented architecture is required.

Figure 6 shows the system which is used for implementing the setup described. The productive system landscape is located in two zones which host the SAP system and the database system. The zones are connected via a virtual private network (VNIC, blue).

This way the communication between the two zones is separated from the public network and does not require further configuration.

The public communication is established via the public virtual interfaces (VNIC, green). As the name implies these interfaces are also virtual and can be configured using the zone configuration tool easily. The network traffic is routed by the physical public interface (NIC). Within the domain name service the virtual hosts of the SAP system and the database are registered via the virtual interface using a virtual hostname. The sense of this setup will become clearer later.

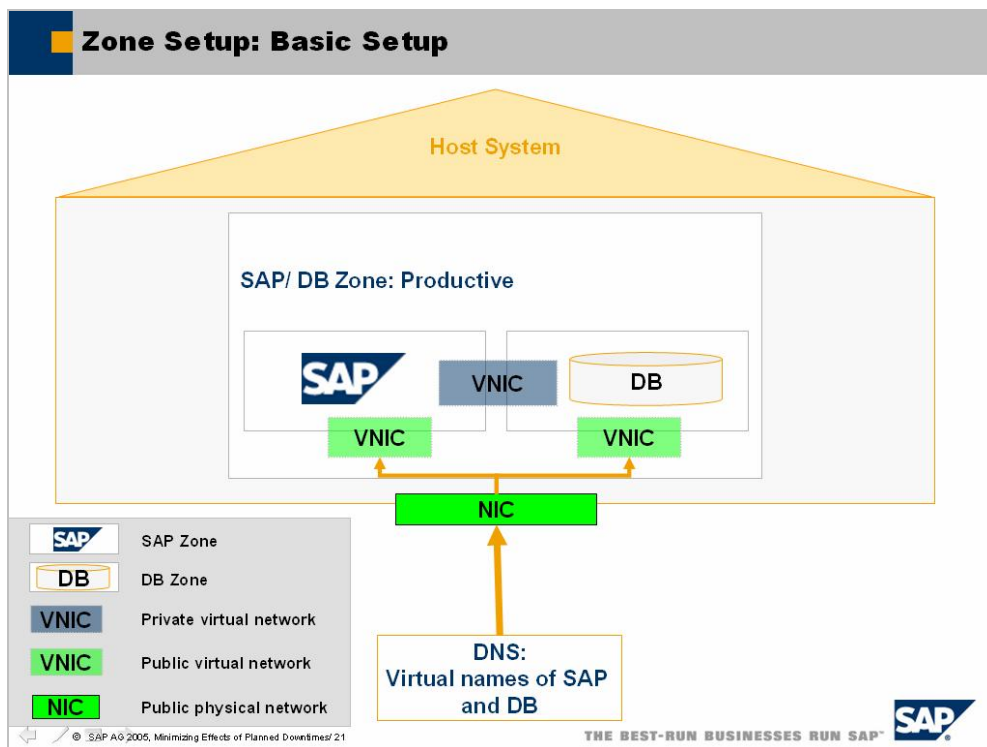


Figure 6: Basic Setup

Based on the system described above a clone of the productive is created as shown in Figure 7. In Solaris this can be done easily using the snapshot/ cloning feature of the Zettabyte File System.

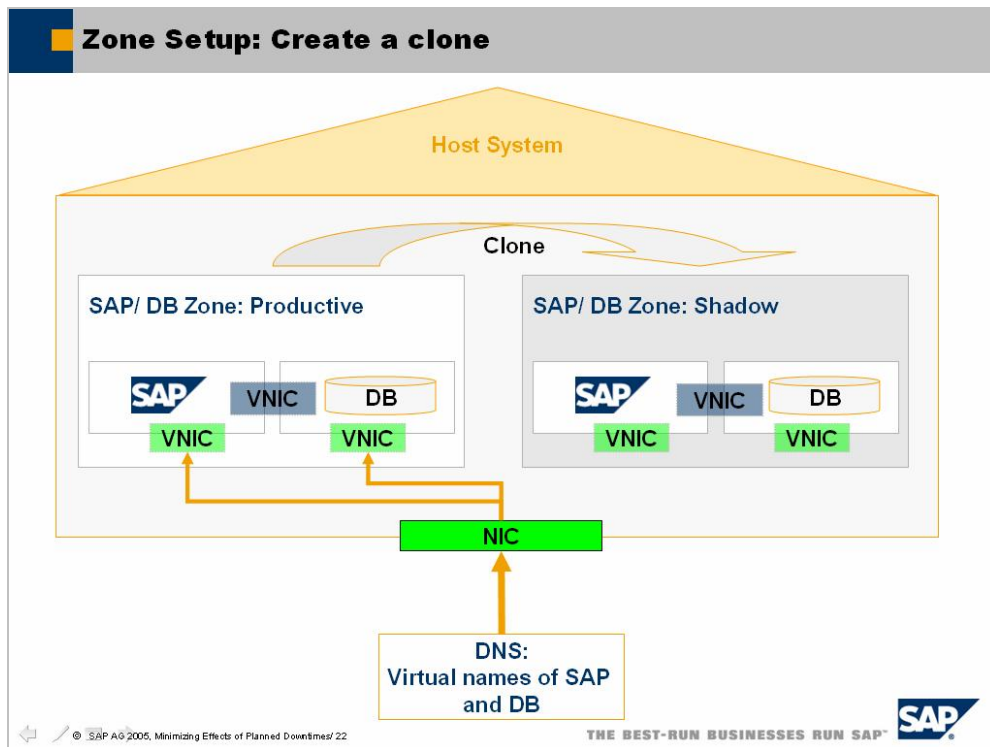


Figure 7: Creating a clone of the SAP system and the database

After cloning the productive zones, the cloned zones can be booted and the shadow systems can be started. Note that the virtual names within the DNS are still bound to the virtual interfaces of the productive system – therefore no downtime occurred yet.

As soon as the clone is created and the systems are started the update can be applied. The easiest way to do this is to install additional public virtual interfaces to access the systems. Of course it is very important to avoid a mix of the productive and the shadow system. For example, it would be fatal if the SAP system installed in the shadow zone communicates to the database installed within the productive zone. Exact procedures for adapting the system after cloning can be found in the SAP System Copy guide.

Also further considerations for managing the access to external systems which are available for the productive system and the shadow system need to be applied.

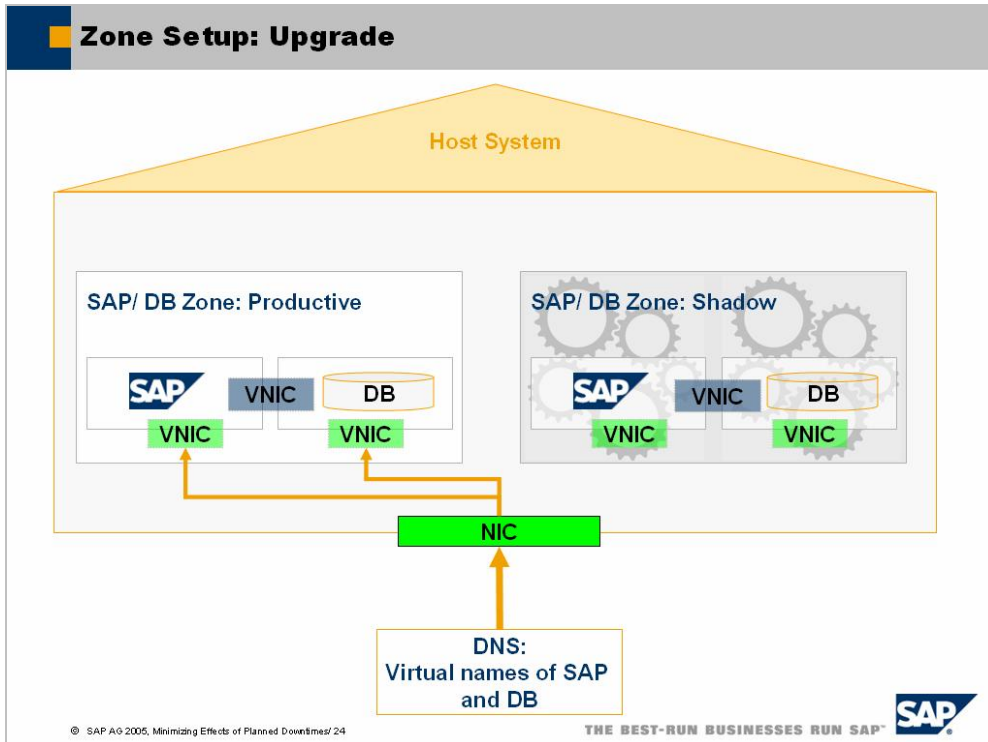


Figure 8: Upgrading the cloned system

If the update and tests succeeded a switch of the systems can be performed. This is reached via a simple network switch: the virtual names of the productive systems registered in the DNS are unbound from the productive system zones and rebound to the shadow system zones – which becomes the new productive system zones afterwards as shown in Figure 9.

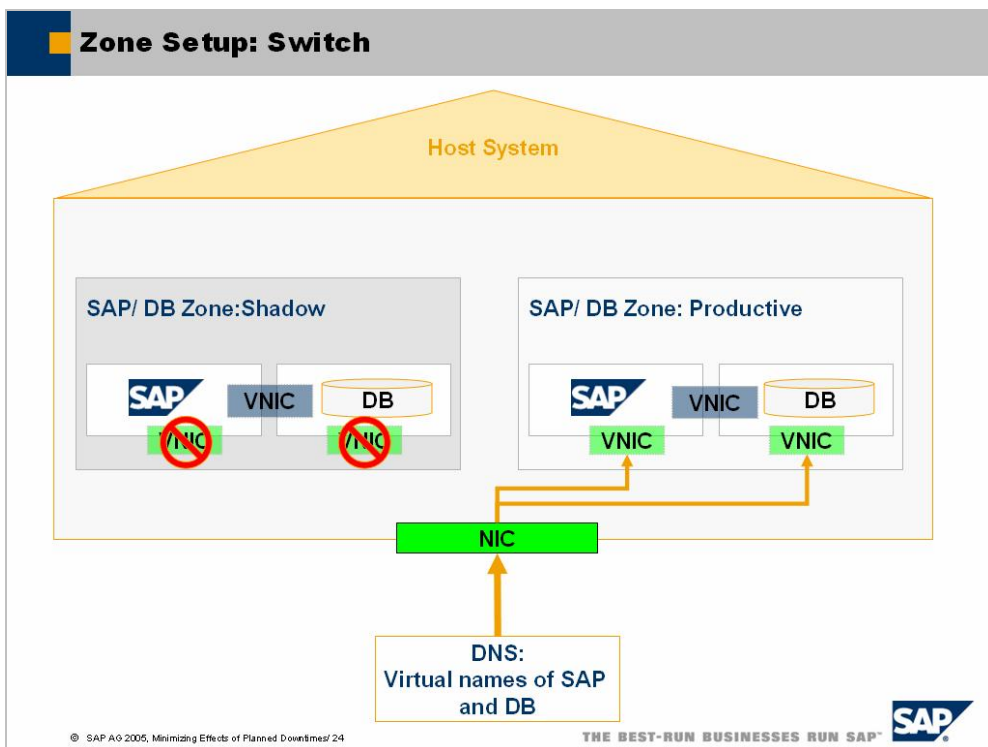


Figure 9: Switching the systems after applying the updates

The advantage of performing the system switch this way is that the effective downtime of the productive system is less than one second usually. In the portal case only the context of the user session would be lost – no new login would be required due to the SAP Logon Ticket is still valid.

Effective Downtime

As described above, the usage of ZFS provides the creation of a snapshot based on the file system within seconds. Therefore it is not necessary to shut down the zone before creating a copy of the productive system as shown in the picture below.

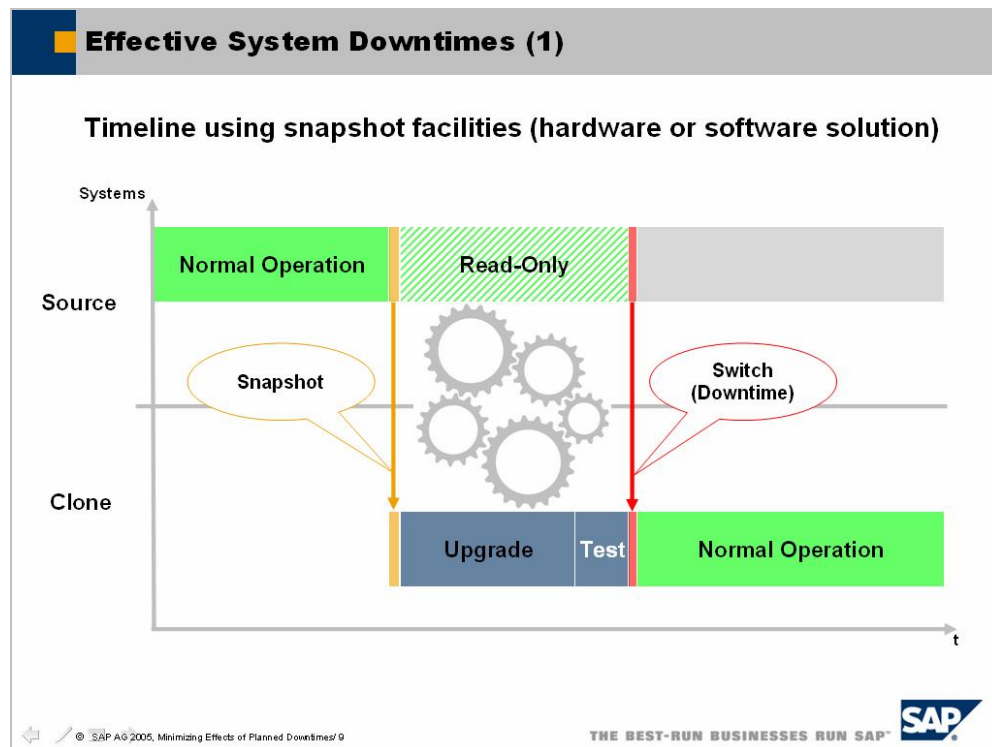


Figure 10: Effective System Downtime using snapshot facilities

Of course, a snapshot could also be performed by using a different solution. For example, a lot of SAN/NAS storage provides these functionalities by default.

Figure 10 summarizes the impact to the productive usage of a SAP system based on the setup described here:

After cloning the system the read-only phase of the productive system begins. This phase ends after the necessary updates and tests have been applied to the shadow system. Afterwards a switch of the network cards is applied as described in Figure 9. Therefore the effective downtime of the system is very short and takes only as long as the switch of the network cards takes. Usually this is done within a second.

Step-by-Step approach

As described above, the setup consists of 3 fundamental steps basically. To provide a more detailed overview the basic steps are listed in this section.

Although the following is Solaris zone specific these steps should also be applicable to many other virtualization techniques. A prerequisite is that the SAP system and the

database are installed already and that resource pools for the productive and the shadow systems are configured already.

■ **Step 1: Creation of the “shadow” zone**

Within this step, the clone of the productive system is created as shown in Figure 7. This is done in online mode using ZFS:

- **Creating a snapshot and a clone:**
ZFS allows to create a snapshot of a file system. Based on that snapshot a clone can be created.
A clone is a kind of writeable snapshot. Technically it is comparable to a delta copy of the source file system: Only the changes from the original file system are saved. Therefore a clone does not require any disk space initially but grows by the changes made within the clone file system.
- **Creating a new zone based on the cloned file system:**
After the clone is create a new zone on top of the clone file system is configured.
- **Adapting zone OS configuration entries:**
To configure a new hostname, host-entries and so on for the zone, some configurations within the zone OS needs to be applied.

■ **Step 2: Upgrade the shadow system**

After the clone is created and booted, the shadow system can be started for applying the update as shown in Figure 8.

■ **Step 3: Switching the systems**

A switch of the system is performed by changing the virtual network interfaces. As shown in Figure 6 both systems have a public virtual interface. The IP's of these interfaces will be switched. Afterwards the cloned system is public available using the virtual address registered in the DNS.

- **Optional - Promoting the clone:**
If system B shall become the new “mastervolume”, promoting the clone is required. For example, this would be necessary if you want to delete the volume of System A afterwards.
To learn more about ZFS features like promoting a clone, consult the online resources as listed in the appendix.
Note: This feature is not included in early versions of ZFS.
- **Adapting zone configuration files:**
To switch the systems in a persistent manner it is necessary to edit the zone configuration files and add the new entries like IP addresses and so on.
- **Optional - Switching some DI's in advance:**
To avoid heavy workload on the primary application server you may decide to switch some of the application servers in advance. Further information on this topic is provided in the “Further Considerations” section.
- **Switching Resource Pools**
Before finally performing the switch of the systems the resource pools should be switched as well to provide the shadow system with the resources reserved for the productive system and vice versa.
- **Switching NIC's and sapmnt share**
When performing the switch of the IP addresses and the sapmnt share System B becomes the productive system and System A becomes the shadow system.
- **Optional - Shut down old system:**
To free resources you may decide to shut down and delete the old system afterwards.

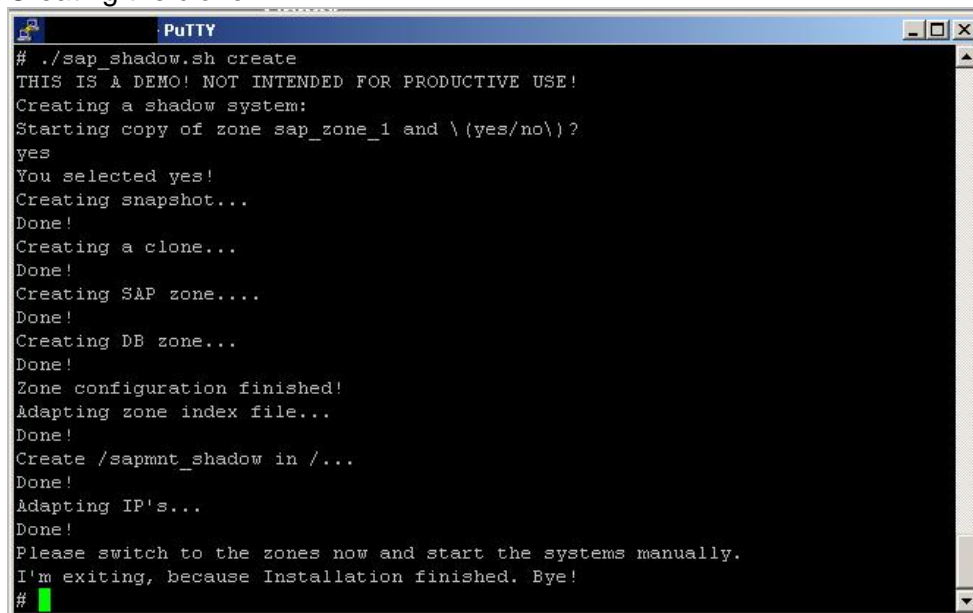
The steps above are only representing a short overview. For detailed information including an extensive step-by-step guide take a look at [xv].

Automated Procedure

To minimize failures and to speed up the process described above, it makes sense to automate this procedure. This can be done via a simple script for example.

For the show case a bash script was created which provides basically two functionalities:

1) Creating the clone

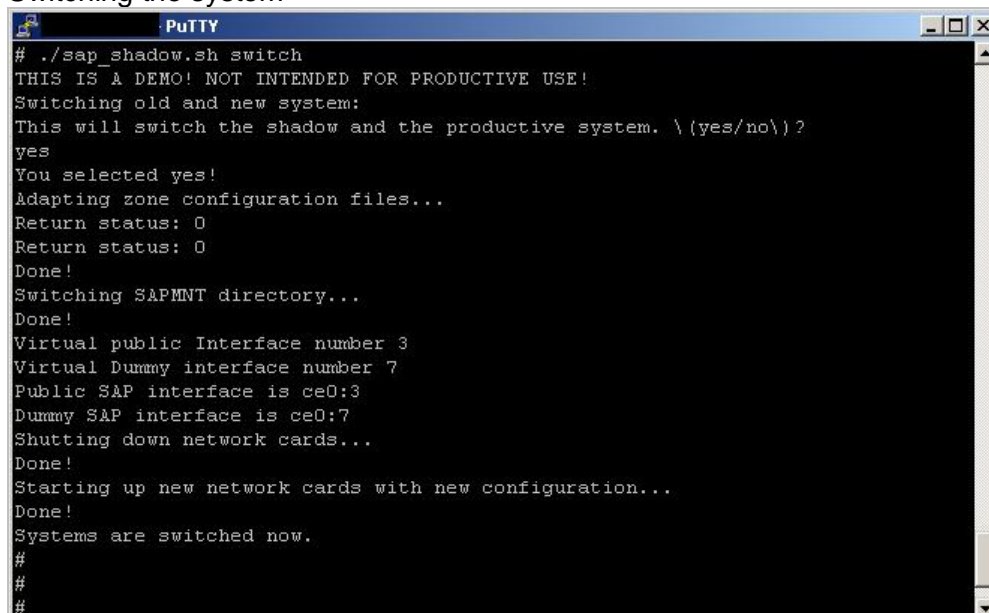


```

# ./sap_shadow.sh create
THIS IS A DEMO! NOT INTENDED FOR PRODUCTIVE USE!
Creating a shadow system:
Starting copy of zone sap_zone_1 and \ (yes/no\)?
yes
You selected yes!
Creating snapshot...
Done!
Creating a clone...
Done!
Creating SAP zone...
Done!
Creating DB zone...
Done!
Zone configuration finished!
Adapting zone index file...
Done!
Create /sapmnt_shadow in /...
Done!
Adapting IP's...
Done!
Please switch to the zones now and start the systems manually.
I'm exiting, because Installation finished. Bye!
#
```

Figure 11: Script to create a clone

2) Switching the system



```

# ./sap_shadow.sh switch
THIS IS A DEMO! NOT INTENDED FOR PRODUCTIVE USE!
Switching old and new system:
This will switch the shadow and the productive system. \ (yes/no\)?
yes
You selected yes!
Adapting zone configuration files...
Return status: 0
Return status: 0
Done!
Switching SAPMNT directory...
Done!
Virtual public Interface number 3
Virtual Dummy interface number 7
Public SAP interface is ce0:3
Dummy SAP interface is ce0:7
Shutting down network cards...
Done!
Starting up new network cards with new configuration...
Done!
Systems are switched now.
#
#
#
```

Figure 12: Script to perform a network switch

Both scripts are executed within a few seconds. The bash script used in the scenario is available at [xvi]. A short demo clip is also available at [xvii]. Please note that this is a bash script written for this specific demo scenario. It is not considered to be a general solution.

Of course, the procedure could be also implemented in any other programming language. Also it could make sense to design the script in a reusable manner to provide these functionalities in regular intervals and on different systems. This way the setup can be performed by experts without special knowledge about the virtualization product implemented.

Quick Review

The steps described above demonstrate how to setup a shadow system using the principles just defined. Table 2 summarizes these principles and the specific implementation in this show case.

Principle	Implementation
Clone	Zettabyte File System/ Solaris Zones
Switch	Virtual Interfaces of Solaris Zones
Isolate	Solaris Zones
Manage Resources	Solaris Resource Management (Containers)
Online Rollback	Zettabyte File System/ Solaris Zones

Table 2: Implementation of Principles

As shown in the table the implementation of the principles may be complementary in many cases. For example, in this show case using Solaris Zones is a way to implement isolation as well as providing a switch mechanism.

Further Considerations

The setup described above only covers the basic steps to create the desired shadow system landscape. Depending of the architecture of system landscape further considerations need to be applied, including:

- **Restart of Dialog Instances:**
After switching the systems all Dialog Instances must be restarted. Due to this would result in very high workload on the primary application server switching some dialog instances in advance may be considered.
Note: SAPCPE must be activated to enable automatic updates of dialog instances.
- **Restrict Access to other systems:**
The access to backend systems should be restricted. Changes on these systems may affect the productive and the shadow system and may cause unpredictable errors.
- **Network Separation**
Access to the shadow system has to be considered carefully: When using the same network as the productive system every system is available for the shadow system, too. This way the shadow system may communicate to other systems that are also available for the productive system.
On the other hand when using a different network system may not be accessible for the shadow system which may affect the operations of the shadow system.
- **Database Caching**
Database Caching (and other caching mechanisms if exist) must be disabled to provide a consistent clone.

- **To be evaluated**

Of course, every specific scenario requires its own considerations. Therefore the list provided above is just an entry point for further evaluation and scenario specific research.

Conclusion

As of today in many productive system landscapes there is still the need to handle with long planned downtime caused by updates, patches and similar tasks. Introducing new techniques and concepts like the shadow system approach can help to minimize the effects of this downtime.

In the use case described above the effective downtime of the productive system is just a second – and therefore not recognized by most of the clients. During the upgrade process the productive system is available in read-only mode. Of course this does not meet every requirement for system availability - but it can help to minimize the effects of planned downtime dramatically.

The concepts described here are should be applicable to the most platforms. For the show case, build in features of Solaris 10 were used and no additional hardware or software products were required. When automating the functionalities in a reusable manner, an update can be applied easily and without deeper knowledge of the environment.

Limitations

The use case described is based on Sun Solaris 10. Of course it cannot be applied 1:1 within different environments.

However, the principles described are representing a general concept and are not related with any specific platform or product. The basic idea is to provide a shadow system which can be updated independently from the productive system and become the new productive system by switching the roles afterwards.

Combining the clone technology with SAP's System Copy also a change of the platform and database is possible. For example after creating a shadow system a system copy could be made based the shadow system. Afterwards this shadow system could be installed on a different host using the same configurations and switched with the productive system later.

However, due to the diversity of system landscapes each setup of a shadow system has to be considered very carefully. Every customer who plans to implement this kind of setup should consult the associated software/ hardware partners for a scenario specific solution therefore.

Outlook

With the upcoming generations of operating systems and hardware architectures new features like ZFS and virtualization will become a standard functionality in many system landscapes.

Large application platforms like SAP NetWeaver can benefit from this trend using these techniques effectively.

For products like the SAP Enterprise Portal a synchronizing mechanism is evaluated. This functionality will allow the creation of a setup described here without any read-only

phase: After cloning and updating the system the data changed in the old system during that timeframe can be synchronized with the new system.

Based on a setup as the shadow system approach described here these new feature would allow to minimize the read-only mode of the productive system drastically.

Appendix

Terminology

System A: The Source	Identifies the “source” system, i.e. the system which exists already before the copy (=clone) is created.
System B: The Clone	Identifies the “cloned” system, i.e. the system that is created within the scenario.
Productive System	Identifies the system which is available to the clients. In Step 1 and 2 this identifies System A. In Step 3 its changed to system B.
Shadow System	Identifies the system which is not available to the clients directly

Tables and Figures

Table 1: System Availability Matrix.....	3
Table 2: Implementation of Principles	18
Figure 1: Planned Downtime Estimations.....	4
Figure 2: Shadow System Approach.....	5
Figure 3: Principles for Minimizing Planned Downtime	6
Figure 4: Sequence of Building Blocks.....	8
Figure 5: Virtualization Levels.....	10
Figure 6: Basic Setup	12
Figure 7: Creating a clone of the SAP system and the database	13
Figure 8: Upgrading the cloned system.....	14
Figure 9: Switching the systems after applying the updates	14
Figure 10: Effective System Downtime using snapshot facilities.....	15
Figure 11: Script to create a clone	17
Figure 12: Script to perform a network switch	17

References

- ⁱ <http://www.hrgresearch.com/pdf/paper4.pdf>
- ⁱⁱ <http://www.hrgresearch.com/pdf/Paper1.pdf>
- ⁱⁱⁱ <http://www.ieeetfcc.org/high-availability.html>
- ^{iv} [https://websmp101.sap-ag.de/%7Eform/sapnet? FRAME=CONTAINER& OBJECT=012003146900000488592004](https://websmp101.sap-ag.de/%7Eform/sapnet?FRAME=CONTAINER&OBJECT=012003146900000488592004) (SMP Logon Required)
- ^v <http://www.kernelthread.com/publications/virtualization/>
- ^{vi} e.g. Limited VMWare support (see SAP Note 674851)
- ^{vii} http://www.sun.com/software/solaris/howtoguides/moving_containers.jsp
- ^{viii} <http://www.vmware.com/products/beta/converter/>
- ^{ix} http://en.wikipedia.org/wiki/Comparison_of_virtual_machines
- ^x http://www.linux-magazine.com/Readers/virtualization/thema/index_html
- ^{xi} http://www.sun.com/software/solaris/containers_learning_center.jsp,
<http://www.opensolaris.org/os/community/zones>
- ^{xii} In Sun's terminology a zone which also uses Solaris resource management capabilities is called a container: http://www.opensolaris.org/os/community/zones/faq/#basic_container
- ^{xiii} http://www.sun.com/software/solaris/zfs_learning_center.jsp,
<http://www.opensolaris.org/os/community/zfs/>
- ^{xiv} <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/4698>
- ^{xv} <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/4804>
- ^{xvi} <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/5066>
- ^{xvii} <https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/5066>

www.sdn.sap.com/irj/sdn/howtoguides