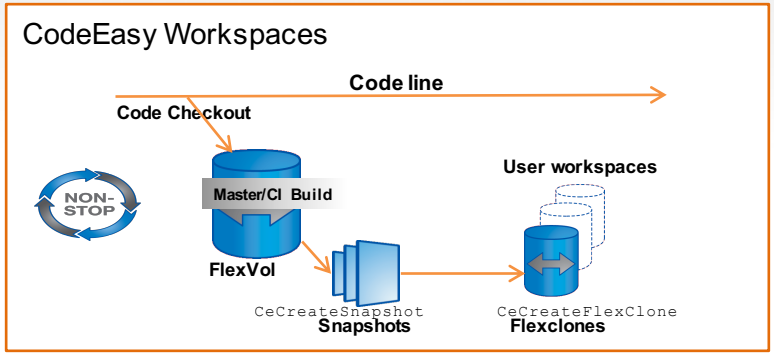
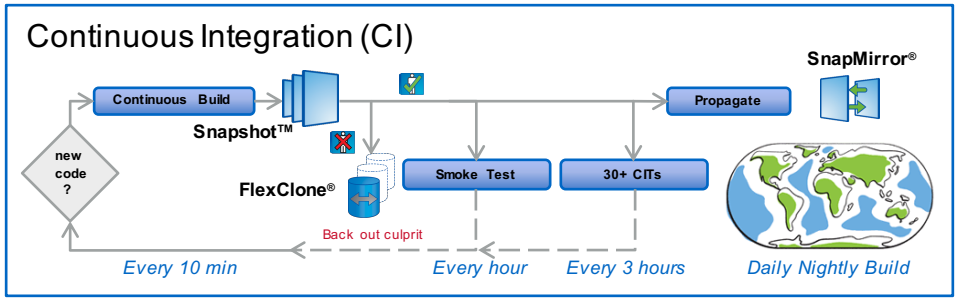
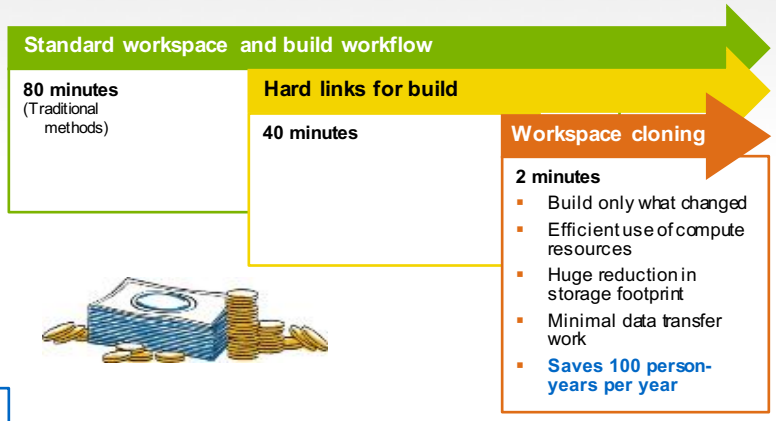
CodeEasy Eval Kit Details

NetApp on NetApp DevOps Story







Key Benefits

- 10x Faster workspaces from SCM—under two minutes for large codebases
- 40x Faster builds with prebuilt object files
- NFS directories for developer sandboxes
- Distributed software development environment
- Continuous integration with minimal storage foot print
- Faster feedback loop
- Keep the code lines stable

NetApp

CodeEasy – Proven Productivity and Efficiency Improvements

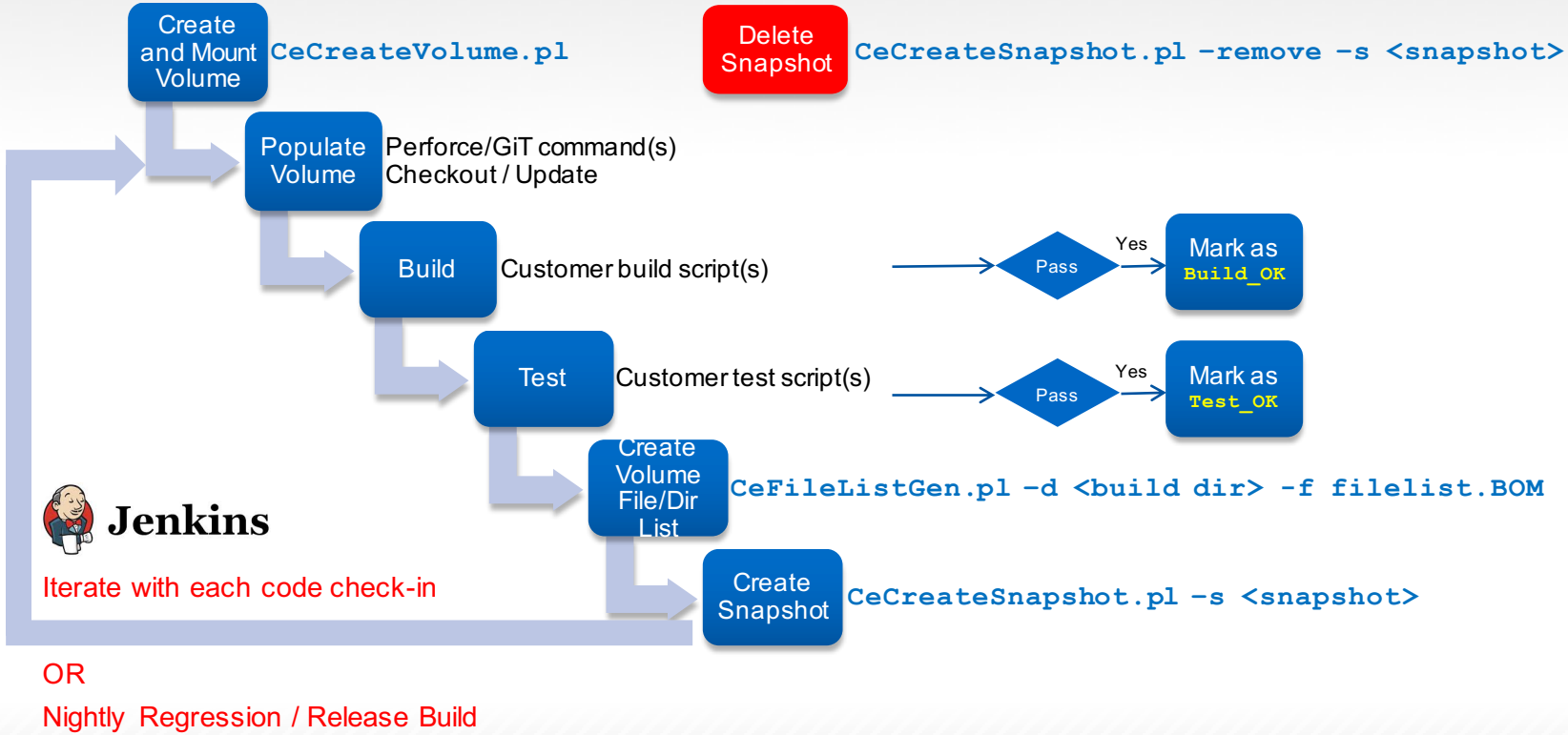
Accelerate design workflows using NetApp Snapshot and FlexClone technology

Customer Enablement

- **CodeEasy Eval Kit**, is a small tarball reference of example scripts and documentation
- Scripts are written in simple, well-documented, easy-to-read Perl
- Scripts utilize **NetApp Manageability SDK** APIs to automate things like create volumes, snapshots and flexclones
- Customers can get CodeEasy working in their environment in **hours**, not days
- CodeEasy can be used with Perforce, GiT, SVN or even CVS

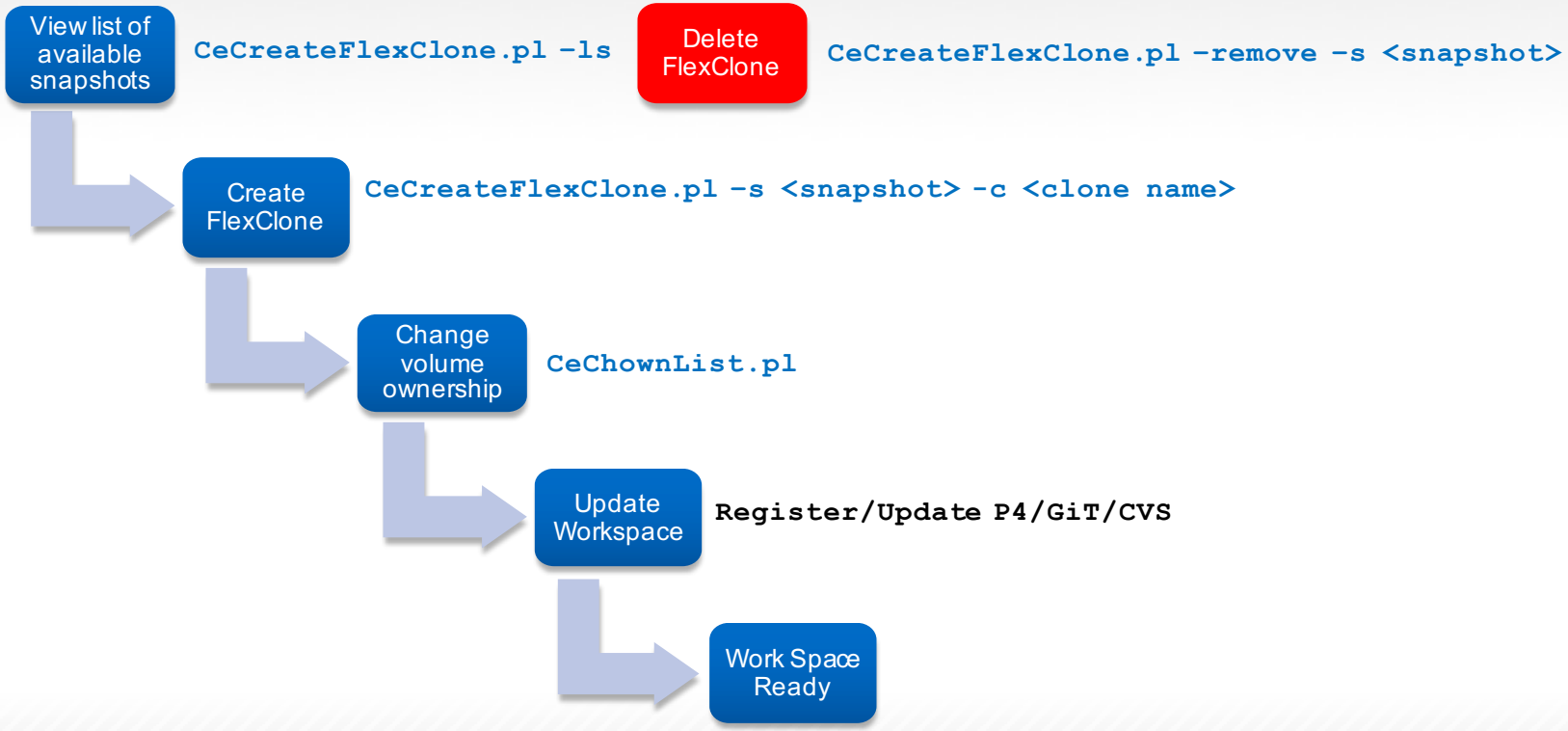
CodeEasy - Build Flow

Iterative Continuous Build/Test/Snapshot flow



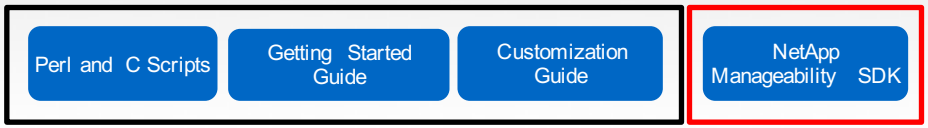
CodeEasy - Developer Workspace Flow

FlexClone Enabled Workspace Creation



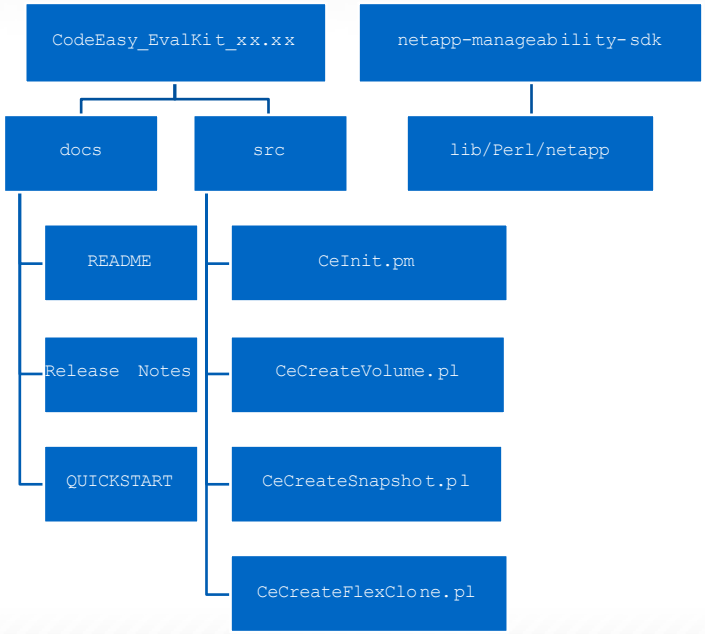
CodeEasy Eval Kit Content

Flow requirements and downloads



- cDOT 8.x
  - The evalkit was developed using cDOT8.2.2.
  - **7-mode is not currently supported**
- CodeEasy_EvalKit_xxxxxx-xxx.tgz
  - Example scripts
- NetApp Manageability SDK 5.4
  - Contains Perl API's (as well as API's for C/C++, Java, MS Net, Python and Ruby)
  - The examples in this eval kit use the Perl API's
  - **This kit must be downloaded by the customer due to EULA requirements**
- Basic knowledge of Perl coding

Eval Kit Structure



Thank You

Additional Slides Content Below

Script Setup and Customization

Flow Permissions

There are three types of users of this flow

- IT/Storage Admin
  - This is the traditional IT storage admin
  - Day to day job is maintaining the storage infrastructure and provisioning storage
  - Will provide sudo access to devops user to do things like chown, etc.
- SW/HW Build Admin (aka user ‘devops’ or ‘build’)
  - Has permissions to create/remove volumes and snapshots
  - Performs regular builds and tests the latest code stream
  - Creates snapshots of the latest builds
- SW/HW Developers
  - Developers who regularly develop and check-in code changes.
  - Will be given access to create a development workspace
  - ‘sur’ will enable user to create/remove FlexClone Volumes

Configuring your storage system

- Configuring your storage system in a cluster environment
  - https://library.netapp.com/ecmdocs/ECMP1200040/html/GUID-A49C52A4-0E67-47C6-96A7-F8BBC055971E.html
- Ensure vserver user access had ontapi application access

```
[STEP 4: CREATE A USER ACCOUNT "rsauser" FOR ONTAPI ON THE CLUSTER VSERVER]
[PASSWORD IS NOT PROMPTED BECAUSE YOU ENTERED ONE IN THE PREVIOUS COMMAND]

clusterab::> security login create -vserver clusterab -username rsauser
-application ontapi -authmethod password -role rsa

[STEP 4: SHOW USER ACCOUNTS]

clusterab::> security login show

Vserver: clusterab
                                Authentication                  Acct
UserName         Application    Method         Role Name        Locked
---------------- -----------    -------------- ---------------- ------
admin            console        password       admin            no
                 http           password       admin            no
                 ontapi         password       admin            no
                 service-processor
                                password       admin            no
                 ssh            password       admin            no
public           snmp           community      readonly         -
rsauser          http           password       rsa              no
                 ontapi         password       rsa              no
Vserver: vs0
                                Authentication                  Acct
UserName         Application    Method         Role Name        Locked
---------------- -----------    -------------- ---------------- ------
vsadmin          http           password       vsadmin          yes
                 ontapi         password       vsadmin          yes
                 ssh            password       vsadmin          yes
11 entries were displayed.
```

Pre-setup

- Create a common user like ‘devops’ or ‘builds’ with special sudo permissions
  - Sudo permissions to run ‘chown’, ‘chmod’ and ‘gcc’

```
# add sudo permissions for user 'devops'
# this will enable user to run command without requiring a password.
devops  ALL=(ALL)       NOPASSWD: ALL
```

- Compile sur.c and fast_chown.c programs
  - These scripts must be compiled as ‘devops’ or ‘root’ since they require special permissions.
  - These scripts are used to changed permissions on the “average” user’s new FlexClone.

Configure Storage Configuration

Storage Admin with help of SW/HW Infrastructure Management

- Configure `CeInit.pm` file
  - Review and edit the entries for the particular code stream
    (project) and Filer configuration as needed
- User Info
- UNIX File System Setup
  - UNIX path for build daemon volumes
  - UNIX path for user FlexClone volumes
- Storage Filer Setup
  - Vserver and aggregate names
  - Junction path locations
  - Volume sizes
  - Volume attributes etc

CeInit.pm (**Update Pic Below**)

```
##########################################################################
# CodeEasy Setup File
#    Basic Initialization Variables
#
# This file should be customized for each project
##########################################################################

# declair this file (.pm) as a Perl package
package CeInit;


########################################
# Users Info - for login permisions etc.
########################################

    our $CE_DEVOPS_USER = "devops";       # user who has filer permissions.  can do any write
                                          # operations including add/remove volume/snaps/etc.
                                          # this is the user the Daemon runs as

    our $CE_USER        = "unknown";      # Average user - set to unknown for now until determined by script
    our $CE_GROUP       = "ubuntu";       # UNIX group name: project or dept group to use

    # NetApp filer access usr/pass pair
    # admin permissions to access filer - used as part of volume creation process
    #   Example:    $naserver->set_admin_user("vsadmin", "devops123");
    our @CE_ADMIN_USER  = ("vsadmin","devops123");
########################################
# NetApp Storage Config Info
########################################
    our $CE_CLUSTER_PORT     = "sv5-devops-01";
    our $CE_DEFAULT_VSERVER  = "sv5-devops-01";
    our $CE_AGGR             = "aggr_devops_05";

    # Storage Mount Points
    # root of the junction path
    # this should be mounted on the unix side, then all other flexclones are automatically mounted when created.
    #       sudo mount -t nfs <vserver>:<junction_path> <unix mount point>
    our $CE_MOUNT_ROOT_DIR    = "/share/devops";

    our $CE_DEFAULT_VOLUME_NAME    = "ce_test_volume";
    # location on the filer where DAEMON's volume and snapshots are stored
    our $CE_MOUNT_DAEMON_ROOT      = "$CE_MOUNT_ROOT_DIR/daemon";
    # location on the filer where USER FlexClone volumes are stored
    our $CE_MOUNT_USER_ROOT        = "$CE_MOUNT_ROOT_DIR/users";

    # Volume attributes
    our $CE_DAEMON_VOL_SIZE        = "3000g";
    our $CE_CLONE_SIZE             = 500 * (10 ** 9);
    our $CE_SSRESERVE_PERCENT      = 20;
    our $CE_VOLUME_SPACE_GUARANTEE = "none";
    our $CE_SSPOLICY_DEVOPS_USER   = "devops_user";
    our $CE_POLICY_EXPORT          = "aws_bb2";
    our $CE_ATIME_UPDATE           = "false";
    our $CE_UNIX_PERMISSIONS       = "775";

    # misc UNIX tool paths - this may need to be modified based on customer
    # environment
    our $CE_CMD_FIND   = "/usr/bin/find";
    our $CE_CMD_XARGS  = "/usr/bin/xargs";
```

CeCreateVolume.pl

Create/Delete Volumes

```
cycrh6svl06.eng.netapp.com:src> ./CeCreateVolume.pl -h

CeCreateVol.pl: Usage Information
      -h|-help                    : show this help info

      -vol|-volume <volume name>  : volume name
                                    default value is set in the CeInit.pm file
                                    by var $CeInit::CE_DEFAULT_VOLUME_NAME

      -r|-remove                  : remove volume

      -v|-verbose                 : enable verbose output

      Examples:
        create a volume named <ce_test_vol>
        %> CeCreateVol.pl -vol ce_test_vol

        remove a volume named <ce_test_vol>
        %> CeCreateVol.pl -vol ce_test_vol -remove
```

- Script is pre-configured for minimal command line input
- This script will be run either by Storage IT or the build user.
  - The average developer will not in this flow create and delete volumes.
  - **NOTE**: This script can be used to create scratch volumes which can dramatically improve the speed of deleting temporary scratch data vs using the '/bin/rm –fr'
- Auto-mounting is truly automatic
  - If automating is enabled and the junction_path root is mounted to the local file system, then any new volumes will be automatically mounted after volume creation.
  - Example – mount the root junction path:

    ```
    %> sudo mount -t nfs sv5-devops-01:/ce_projects /ce_projects
    ```

    If you create a new volume '`regressions`' the volume would automatically show up at UNIX path `/ce_projects/project_A/regressions`

CeFileListGen.pl

Very fast full volume file/directory list generator

- Creates a inventory list of all the files and directories in the volume.
- The list is used to speed-up the change of ownership process after creating a FlexClone.

```
------------------------------------------------------------
CeFileListGen.pl: file list generator
------------------------------------------------------------

CeFileListGen.pl: Usage Information
      -h|-help                          : show this help info

      -d|-directory <directory name>    : root directory to scan
                                          the program will inventory all files
                                          and directories from this directory
                                          downward. use full UNIX path
      -f|-filelist  <filelist name>     : name of the filelist to generate
                                          default=<directory name>/filelist.BOM
                                          (optional)

      -v|-verbose                       : enable verbose output

      Examples:
        create a filelist called /my_path/dir_to_scan/filelist.BOM
        %> CeFileListGen.pl -d /my_path/dir_to_scan/
```

- The script was designed to handle large volumes efficiently.
- The level of parallelism used can be tuned to the environment.

NOTE: run this script prior to creating a Snapshot, so the file list is included in the FlexClone image.

CeCreateSnapshot.pl

Create/Delete Snapshots

```
cycrh6svl06.eng.netapp.com:src> ./CeCreateSnapshot.pl -h

CeCreateSnapshot.pl: Usage Information
      -h|-help                        : show this help info

      -vol|-volume   <volume name>    : volume name
                                        default value is set in the CeInit.pm file
                                        by var $CeInit::CE_DEFAULT_VOLUME_NAME

      -s|-snapshot <snapshot name>    : volume name

      -r|-remove                      : remove snapshot

      -v|-verbose                     : enable verbose output

      Examples:
        create a snapshot for volume ce_test_vol
        %> CeCreateSnapshot.pl -volume ce_test_vol -snapshot ce_test_vol_snapshot_01
```

- Suggested Naming Conventions
  - <snapshot>_<build number>
- Importance of a naming convention
  - A naming convention will provide a meaningful connection between the SCM version and the snapshot.
  - Users will reference the Snapshot name when creating a FlexClone
- Mount Location
  - UNIX: `<mount>/<volume>/.snapshot/<snapshot name>`

CeCreateFlexClone.pl

Create developer workspace from an existing Snapshot

```
CeCreateFlexClone.pl: Usage Information
      -h|-help                      : show this help info

      -vol|-volume  <volume name>   : volume name
                                      default value is ce_test_vol
      -s |-snapshot <snapshot name> : name of the snapshot to clone
      -cl|-clone    <clone name>    : name of the snapshot to clone
      -c|-create                    : create volume
      -r|-remove                    : remove volume

      -v|-verbose                   : enable verbose output

      Examples:
        create a FlexClone with the name <ce_test_vol>
        starting with snapshot
        %> CeCreateFlexClone.pl -volume ce_test_vol -snapshot ce_test_snapshot
                    -clone  my_flexclone_ce_test -create
```

- Clones are based on existing volume snapshots
- Suggested Naming Conventions
  - <snapshot>_clone
- Mount Location
  - Filer: **<junct_path>**/users/<username>/FlexClone name>
  - UNIX: **<mount>**/users/<username>/<FlexClone name>
- Auto-mounting is truly automatic
  - If junction paths are used.

FlexClone Storage Management

- **List the FlexClones**

```
%> CeCreateFlexClone.pl -lc
```

```
INFO  (CeCreateFlexClone.pl): Connecting to storage controler/vserver
        storage controler  = sv5-devops-01
        set_vserver        = sv5-devops-01
        set_transport_type = HTTP
INFO  (CeCreateFlexClone.pl): Storage Controller <sv5-devops-01> is running ONTAP API version:
      NetApp Release 8.2.1RC2X6 Cluster-Mode: Wed Dec 18 19:14:04 PST 2013


List FlexClones
Parent Volume            Parent-Snapshot              FlexClone                       Parent Vol  FlexClone Vol     Split Est         FlexClone Act    Cloan Owner  Junction-path
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
viper_nightly_builds     nightly_20150416_1626        nightly_20150416_1626_clone1     447.21 MB      451.37 MB      428.91 MB      22.45 MB ( 4.97%)       jmichael  /proj/viper/user
viper_nightly_builds     nightly_20150416_1626        nightly_20150416_1626_clone2     447.21 MB      446.34 MB      426.53 MB      19.81 MB ( 4.44%)       donjulio  /proj/viper/user
viper_nightly_builds     nightly_20150416_1637        nightly_20150416_1637_clone1     447.21 MB      447.23 MB      427.51 MB      19.71 MB ( 4.41%)       donjulio  /proj/viper/user
viper_nightly_builds     nightly_20150416_1637        nightly_20150416_1637_clone2     447.21 MB      893.50 MB      423.38 MB     470.12 MB (52.62%)     josecuervo  /proj/viper/user
viper_nightly_builds     nightly_20150416             nightly_20150416_clone1          447.21 MB      446.31 MB      426.77 MB      19.54 MB ( 4.38%)       jmichael  /proj/viper/user
viper_nightly_builds     nightly_20150416             nightly_20150416_clone3          447.21 MB      446.31 MB      426.91 MB      19.39 MB ( 4.35%)      cptmorgan  /proj/viper/user

CeCreateFlexClone.pl exited successfully.
```

- **FlexClone Actual Size = FlexClone Vol Size – Split Est Size**
  - Where the “Split Estimate” is the amount of shared storage between the clone and its parent volume.
- **FlexClones consume <5% of the full volume**

CeChownList.pl

Script to change file/dir ownership

```
CeChownList.pl: Usage Information
      -h|-help                          : show this help info

      -d|-directory <directory name>    : root directory to chmod

      -f|-filelist  <filelist name>     : name of the filelist to read
                                          default=<directory name>/filelist_BOM
                                          (optional)

      -u|-user <username>               : username for chown command

      -v|-verbose                       : enable verbose output

      Examples:
        create a filelist called /my_path/dir_to_scan/filelist.BOM
        %> CeChownList.pl -user user_larry -d /my_path/dir_to_chmod/ -f /my_path/filelist_BOM
```

- Why is this needed?
  - FlexClones retain the file and directory permissions of their parent Snapshot.
  - Ownership must be changed to the developer
- Very fast script for changing ownership of files/volumes.
  - Uses parallel processing to speed up performance
  - CeChownList.pl reads the file/vol list created by CeFileListGen.pl prior to creating the snapshot.
  - fast_chown (fast_chown.c) is called by CeChownList.pl
    - fast_chown.c uses the lchown() function to speed up changes to file and directory user:group ownership.
    - `%> fast_chown`
      `Usage: %> fast_chown <user id> <file list>`
- Permissions to change file/director ownership
  - sur.c is provided in the kit, which when compiled as root (or sudo), can switch uid and then run a sub process/script as the designated user.
  - Example:

```
Usage:  %> ./sur <username> <sur_dispatch cmd> <cmdline to execute>

        %> ./sur devops ./CeChownList.pl -d /proj/users/ce_test_snap_01_clone -u larry
```

REMEMBER

This process requires sudo permissions.

Perforce Commands

Register new FlexClone volume with Perforce

- Assumption
  - Perforce environment is setup and available.
- After creating a FlexClone workspace
  - %> cd <new FlexClone directory>
  - Step 1: Define P4 Client workspace
    - Define a Perforce client to let Perforce know the existence of the new workspace, modifying fields such as Root directory and Owner with user-specific information.
    - Run %> p4 client
  - Step 2: Run p4 flush
    - Run '**`%> p4 flush`**' to make the workspace think it has the file content already.

GiT or SubVersion (SVN) Commands

- After creating a FlexClone workspace
  - Nothing needs to be done except for changing the file ownership.
  - GiT and SubVersion (SVN) store no users specific meta data in the local workspace – so no changes required.

CVS Commands

Requires Testing and Verification

- After creating a FlexClone workspace
- Change the ownership of the files in the CVS directories

```
%> find . -name Root -type f | xargs -P 140 perl -pi -e 's/old_user/new_user/g'
```

Suggested Customization

Customization considerations

- Create standard for volume, snapshot and flexclone names
- Junction Points and Mount Paths
  - Naming/storage conventions making managing the volumes, snapshots and FlexClones easier.
  - Conventions used in CodeEasy…
- Use links to map to user areas to FlexClones
  - In my homedir:
    /u/anissam/p4/mybb

    will symlink to:
    /x/eng/bbsvl/users/anissam/SVL_mybb_750259_01101204
- Process/script tracking Snapshots and “quality” of Snapshots
  - Users would query the available snapshot to determine the snapshot they want to Clone.

Customization considerations

- Define policies around max Snapshots to retain
- Define policies around max FlexClones per user
- Take filer loading into account when creating new volume
  - Pick the least loaded controller

Recommendations

Recommendations and Best Practices

CodeEasy Best Practices

WORK IN PROGRESS

- Recommendations
  - Max FlexClones per user – 15
  - Max Snapshots – 150/daemon volume?
- cDOT 8.1
  - Max FlexClones per Snapshot/Volume: 32,767
  - Max volumes per cluster: 1200
  - Max volumes per controller: 1000
- Directory structure recommendations etc.

Directory structure / Layout

```
Project Directory Structure (UNIX View)
=======================
ce_projects
├── project_A
├── project_B
└── project_C
    ├── jenkin_build
    │   └── <project contents>
    ├── nightly_regression
    ├── release_builds
    └── users
        ├── cptmorgan
        │   ├── project_C_snap2_clone
        │   ├── project_C_snap3_clone
        │   └── project_C_snap4_clone
        │       └── <project contents>
        ├── jackdaniels
        │   ├── project_C_snap2_clone
        │   ├── project_C_snap5_clone
        │   └── project_C_snap6_clone
        ├── jameson
        │   ├── project_C_snap3_clone
        │   └── project_C_snap4_clone
        └── josecuervo
            ├── project_C_snap1_clone
            ├── project_C_snap2_clone
            └── project_C_snap3_clone
```

Filer View	Junction Path
ce_projects	/ce_projects
project_A (qtree)	/ce_projects/project_A
project_B (qtree)	/ce_projects/project_B
project_C (qtree)	/ce_projects/project_C
project_C_jenkin_build	/ce_projects/project_C/jenkin_build
<project files/directories>	
project_C_nightly_regression	/ce_projects/project_C/nightly_regression
project_C_release_builds	/ce_projects/project_C/release_builds
qtree or UNIX dir	qtree or UNIX dir
qtree or UNIX dir	qtree or UNIX dir
project_C_snap2_clone	/ce_projects/project_C/users/cptmorgan/project_C_snap2_clone
...	...
project_C_snap4_clone	/ce_projects/project_C/users/cptmorgan/project_C_snap4_clone
<cloned version of project files/directories>	
...	

CodeEasy FAQ

Frequently Asked Questions – with Answers…

FAQ's

- How do you handle the 255 snapshot limit per volume when we have thousands of developers?
  - Create a Flexclone off of a Snasphot and move it to the least loaded aggregate, so that developer can create up to 32767 FlexClones off a FlexClone. Please find the attached of BammBamm on internal arch design to handle this.
  - https://wikid.netapp.com/w/Bammbamm_Shadow_Volume_Testing
  - http://wikid.netapp.com/uploads/e/ed/BammBamm_WhiteSession.pptx
- Does the volume limit pose issues if I have many developers who want to make many clones?
  - Two guidelines:
    - Define the volume purge policy on limiting the number of workspace and deleting them automatically, if there any not active by looking into the SCM opened file stamps or build time stampes
    - Add more nodes to the cluster to get more volumes and develop the tools, which can work agnostic to underlying junction path.
- Workspaces live for a long time, even years. Will this cause snapshots to be locked for that period of time? Over time these workspaces will diverge considerably from the original clone – does this mean that the overall value diminishes?
  - Volumes can be moved to another aggregate hosted on cheaper storage (SATA disks) . Design the cluster to support multitier performance requirements
- Will having too many clones on one controller cause too much load to be on that filer? How can I manage that?
  - It should not be. No known issues
- Can this process be implemented in 7-mode?
  - The initial implementation works with cDOT. Investigation into a 7-mode implementation will be done based on key customer demand.

FAQ's

- Can this process work with ClearCase?
  - May work for Snapshot views and does not work for Clearcase dynamic views
- Does having all these FlexClones make the workspaces slower for file access than if the files were "real"?
  - Not Really
- How do I manage all of these FlexClones? How do I know when I can delete them?
  - This is goes back to Software development and Continuous integration process at each company. There should be automated process defining the life cycle of the workspace with a purge policy
- How does this process work in a multi-site scenario?
  - Every site would require similar setup and the tools and other configuration setup can be snap-mirrored to keep the environment consistent.

Use of FlexClones

- In environments that make heavy use of FlexClones, the overall performance increases.
  - That's because multiple FlexClones working off the same set of 4KB blocks 'effectively' allow more 4KB blocks to fit in cache and main memory.
  - FlexClones have the effect of making FlexCache and main memory 'bigger'.
  - You see this mostly in VMware environments, where they make heavy use of FlexClones (& de-duplication). Once you eliminate the churn of running multiple VM's with almost the same identical set of 4KB blocks thru the same main memory, the system typically runs faster - as you're eliminating the iOPs to HDD.

- Part of the attraction of cDOT in this environment is the flexibility to move FlexVols/FlexClones from 1 node to another in a cluster when the individual node gets overloaded. It's typically only a matter of time in these environments as to when the a node gets oversubscribed.