Care Coordination

Decision Support Tool (DST)

Deployment, Installation, Back-Out, and Rollback Guide

May 2019

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Revision History

Date	Version	Description	Author
05/15/2019	0.4	Defect remediation	AbleVets LLC
04/26/2019	0.3	Defect remediation	AbleVets LLC
04/24/2019	0.2	Added TLS installation specific to GMRC*3.0*125 Patch Information	AbleVets LLC
04/11/2019	0.1	Initial Draft	AbleVets LLC

Artifact Rationale

This document describes the Deployment, Installation, Back-out, and Rollback Plan for new products going into the VA Enterprise. The plan includes information about system support, issue tracking, escalation processes, and roles and responsibilities involved in all those activities. Its purpose is to provide clients, stakeholders, and support personnel with a smooth transition to the new product or software, and should be structured appropriately, to reflect particulars of these procedures at a single or at multiple locations.

Per the Veteran-focused Integrated Process (VIP) Guide, the Deployment, Installation, Back-out, and Rollback Plan is required to be completed prior to Critical Decision Point #2 (CD #2), with the expectation that it will be updated throughout the lifecycle of the project for each build, as needed.

Table of Contents

List of Tables

1. Introduction

This document describes how to deploy and install the Community Care Decision Support Tool (DST) as well as how to back-out the product and rollback to a previous version or data set if applicable. This document is a companion to the project charter and management plan for this effort. This document details the criteria for determining if a back-out is necessary, the authority for making that decision, the order in which installed components will be backed out, the risks and criteria for a rollback, and authority for acceptance or rejection of the risks.

1.1 Purpose

The purpose of this plan is to provide a single, common document that describes how, when, where, and to whom the DST be deployed and installed, as well as how it is to be backed out and rolled back, if necessary. The plan also identifies resources, communications plan, and rollout schedule. Specific instructions for installation, back-out, and rollback are included in this document.

1.2 Dependencies

The DST Application is dependent on the following Systems/Applications/Services.

Table 1: DST Application Dependencies

Dependency	Type	Dependency Type	DST Use
Computerized Patient Record System (CPRS)	System	System	Consult data is supplied to DST. This data is used to initiate a DST decision for a given consult. Detailed data fields are shown within Section 7 – External Interfaces in this document.
Master Veteran Index (MVI)	Service	Data/Information	Internal data service to access MVI external data. Will contain all unique query logic to interact with the external service, along with external interface configuration setup (such as authentication).
Corporate Data Warehouse (CDW)	Service	Data/Information	Internal data service to interact and query CDW cached data. Data will be a scheduled task to load CDW into the DST environment. CDW data will reside within DST for lookup and reference within the DST decision logic. The data will have its own designated datastore due it being relational data.

Dependency	Type	Dependency Type	DST Use
Enrollment System Redesign (ESR)	Service	Data/Information	Internal data service to access Enrollment Service external data. Will contain all unique query logic to interact with the external service, along with external interface configuration setup (such as authentication).
Provider Profile Management System (PPMS)	Service	Data/Information	Internal data service to access PPMS external data. Will contain all unique query logic to interact with the external service, along with external interface configuration setup (such as authentication).
Standardized Episodes of Care (SEOC)	Service	Data/Information	Internal data service to access SEOC stored internal data. Will contain all unique query logic to interact with the datastore to query data, including configuration setup (such as authentication).

1.3 Constraints

The DST project team, software, and test servers will adhere to the following directives, policies, procedures, standards, and guidelines:

- Veteran-focused Integration Process (VIP).
- Section 508 Information Technology (IT) accessibility standards governed under 29 U.S.C 794d.
- Health Insurance Portability and Accountability Act (HIPAA).
- VA DIRECTIVE 6508 - Privacy Impact Assessments.
- VA Directive 6500 – Information Security Program.
- One-VA Technical Reference Model (TRM).
- VA Standards & Conventions Committee (SACC) Codes Standards and Conventions.
- The DST will pass any WASA scans.
- The DST will not have any Critical or High issues identified by a Fortify scan.

2. Roles and Responsibilities

Please refer to the following table for the deployment, installation, back-out, and rollback roles and responsibilities.

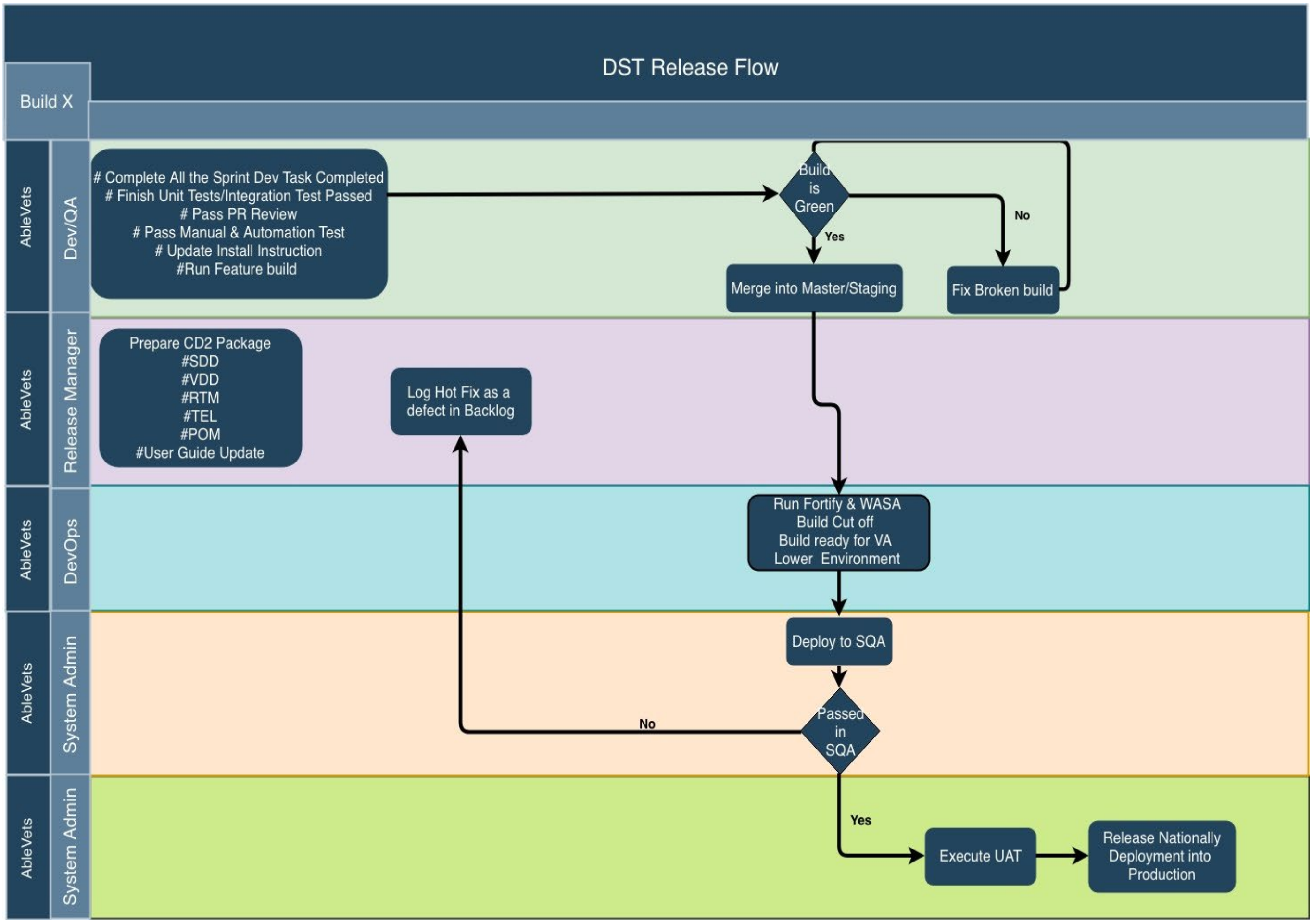
Table 2: Deployment, Installation, Back-out, and Rollback Roles and Responsibilities

ID	Team	Phase / Role	Tasks
1	AbleVets Development	Deployment in Local Dev	Plan and schedule deployment in local environment
2	AbleVets DevOps Team	Deployment in Software Quality Assurance (SQA)/User Acceptance Testing (UAT) in Department of Veterans Affairs (VA)	Determine and document the roles and responsibilities of those involved in the deployment.
3	AbleVets DevOps	Deployment in Production	Test for operational readiness
4	AbleVets DevOps	Installation	Plan and schedule installation
6	VA	Installation	Validate through facility POC to ensure that IT equipment has been accepted using asset inventory processes
8	AbleVets DevOps	Back-out	Confirm availability of back-out instructions and back-out strategy (what are the criteria that trigger a back-out)
9	AbleVets DevOps	Post Deployment	Hardware, Software and System Support

3. Deployment

The deployment is planned as an iterative rollout. The following swim lane provides the high-level overview of DST Release Process.

Figure 1: Overview of the DST Release Process



3.1 Timeline

This section providers the project schedule and milestones.

Table 3: DST Task Names and Start Dates

Task Name	Start Date	End Date
Deploy DST into Software Quality Assurance (SQA)	04/01/2019	04/01/2019
Testing in SQA	04/05/2019	04/24/2019
User Acceptance Testing (UAT)	04/23/2019	04/26/2019
Initial Operating Capability (IOC) Testing	05/08/2019	05/14/2019
Deploy DST into Production	05/07/2019	06/03/2019
Deploy Consult Toolbox (CTB) into Production	05/09/2019	05/31/2019

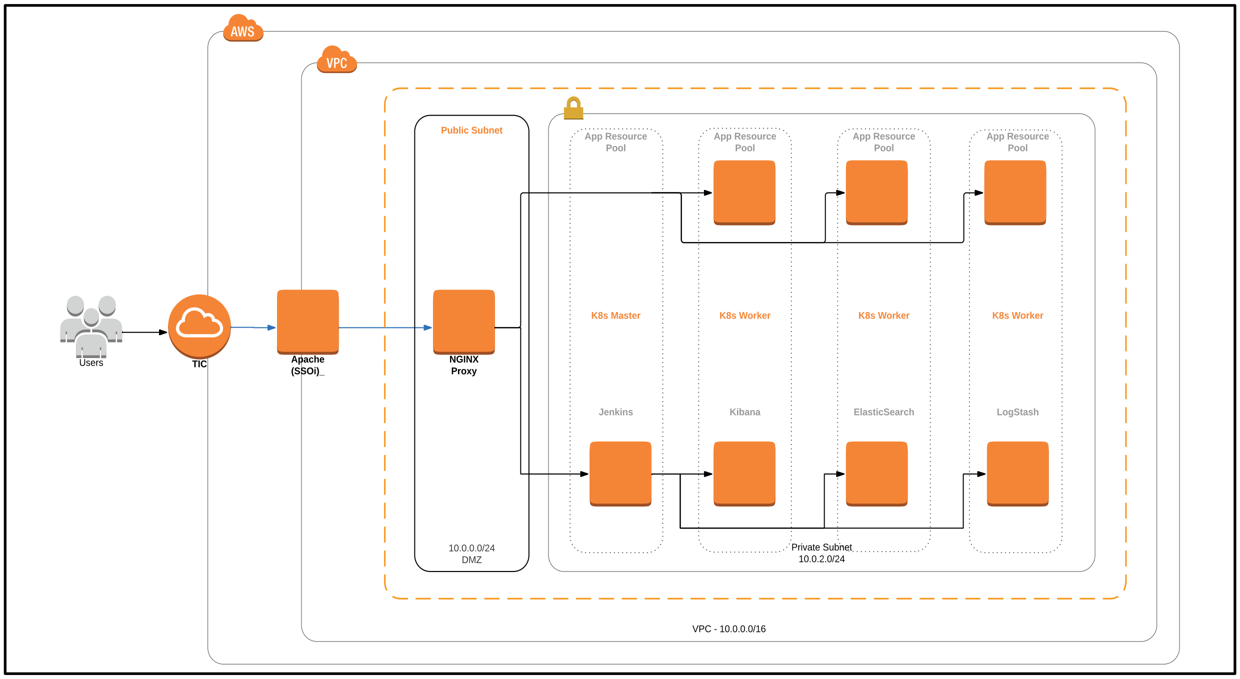
Task Name	Start Date	End Date
Deploy VistA Patch into Production	05/24/2019	05/29/2019

3.2 Site Readiness Assessment

The DST application will exist within the VA Enterprise Cloud for DEV, PREPROD, and Production environments. The DST development team will maintain a local DEV to be used for sprint development and testing processes.

3.2.1 Deployment Topology (Targeted Architecture)

Figure 2: Deployment Topology (Targeted Architecture)



3.2.2 Site Information (Locations, Deployment Recipients)

The initial deployment of the DST web interface will be to IOC sites so that users can verify the functionalities of DST. Once testing is completed and DST is approved for national release, DST will be deployed nationally.

DST will be deployed to the following IOC sites.

- Anchorage AK
- Madison, WI
- Philadelphia, PA
- Salisbury, NC
- Kansas City, MO
- Ft. Harrison, MT

3.3 Resources

This section describes hardware, software, facilities, documentation, and any other resources, other than personnel, required for deployment and installation.

3.3.1 Hardware

DST is in the VAEC cloud enclave. There are four cloud environments maintained. All environments have a common hardware parity with the hardware specifications listed below. All application software and microservice configuration (Kubernetes) are executed on the hardware.

Please refer to Table 2 in the Roles and Responsibilities section of this document for details about who is responsible for preparing the site to meet these hardware specifications.

Figure 3: Hardware Resources

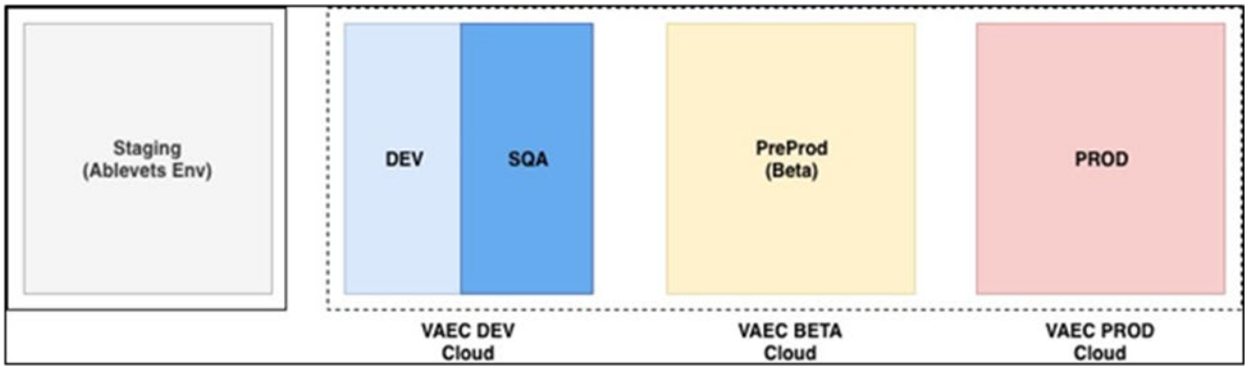


Table 4: Hardware Specifications

Required Hardware	Model	Version	Configuration	Manufacturer	Other
AWS	M5	Large	Virtual	Virtual	All Servers

Technology Component Production 1	Location	Usage
DST Production – VA Cloud	VA Cloud environment	To serve the DST application within the VA Production environment.

Technology Component Verification/Test	Location	Usage
DST PreProd – VA Cloud	VA Cloud environment	To test the DST application within a VA test and/or verification environment.

Technology Component Verification/Test	Location	Usage
DST DEV/SQA – VA Cloud	VA Cloud environment	To test the DST application within a VA test and/or verification environment.

Technology Component Development	Location	Usage
DST Development – AbleVets Cloud	Ablevets Cloud environment	To develop, test, and demo the DST application before transition to the VA cloud environment.

3.3.2 Software

The following table describes software specifications required prior to deployment. If there are difference depending upon site, those difference will need to be provided.

Table 5: Software Specifications

Required Software	Make	Version
Apache	Apache Software	2.4.X
Kubernetes	Red Hat	1.13.X
Docker	Docker, Inc	18.06.0-ce
Red Hat	Enterprise Linux Server	7.X

Please refer to Table 2 in the Roles and Responsibilities section of this document for details about who is responsible for preparing the site to meet these software specifications.

3.3.3 Communications

Notification of scheduled maintenance periods that require the service to be offline or that may degrade system performance will be disseminated to the business user community a minimum of 48 hours prior to the scheduled event.

Notification to VA users for unscheduled system outages or other events that impact the response time will be distributed within 30 minutes of the occurrence.

Notification to VA users for unexpected system outages or other events that impact the response time will be distributed to Users as soon as possible.

Notification will be distributed to VA users regarding technical help desk support for obtaining assistance with receiving and processing.

3.4 Deployment/Installation/Back-Out Checklist

The table below outlines the coordination effort and documents for the day/time/individual when each activity (deploy, install, back-out) is completed for DST.

Table 6: Deployment/Installation/Back-Out Checklist

Activity	Day	Time	Individual who completed task
Deploy	TBD	TBD	TBD
Install	TBD	TBD	TBD
Back-Out	TBD	TBD	TBD

4. Installation

4.1 Platform Installation and Preparation in Facility level

DST requires the following three separate components to be deployed into Production for each facility level.

4.1.1 Consult Toolbox 1.9.0004

CTB 1.9.0004 is the AutoHotkey component of the DST solution that is installed as a thick-client on the CPRS user's workstations by the VA-ITOPS team. It is responsible for monitoring the state of which CPRS screen is displayed to the User, presenting the user with the option to launch DST, and facilitating the transfer of information between CPRS and the DST API/Database. The deployment of CTB 1.9.0004 includes a dedicated DST .ini file that includes a string parameter containing the root URL for the DST endpoints. When this parameter is NULL or the DST .ini file is not found, Consult Toolbox does not attempt any communication with DST and operates based on its pre-DST user experience. The initial national deployment of CTB 1.9.0004 will be deployed with the DST URL set to an empty string.

During Quality Assurance and User Acceptance testing, CTB 1.9.0004 should be tested in configurations with and without the ini file, with and without the DST URL parameter, to insure proper operation in all modes of operation. Multiple test DST URLs will be provided to allow comprehensive testing of error states, edge cases, and all possible DST use cases. These changes have all been tested in DEV, SQA, UAT, and IOC testing events.

4.1.2 VistA DST Patch to the GMRC Package at Each VistA Site

DST VistA Patch GMRC*3.0*124 is the Vista component of DST which must be installed on every VistA system whose CPRS users need to use DST. The patch includes a protocol that invokes a process to retrieve the consult factor text from DST and insert it into a consult comment whenever a consult is signed that contains the string "DST ID:" in the Reason for Request field. Until DST becomes operational in production, the only way that the DST process would be invoked would be if someone manually added the "DST ID:" string to a consult. If this occurs prior to the deployment of the DST endpoint, the VistA process will receive an HTTP error and will add a consult comment with text indicating that the DST Service was not available (useful for testing, but not disruptive to normal operation). The Patch will essentially lie dormant until the first UAT/IOC testing in Production starts when actual DST ID Global Unique Identifier (GUIDs) are first written to consult orders.

If the DST URL is not active during SQA testing, a test endpoint will be created to allow for end-to-end testing of the DST patch operation. Detect “DST ID:”, retrieve consult factor text from DST API, and insert the consult factor text into a newly created comment.

4.2 Creating SSL/TLS Encryption for VistA to DST Communication

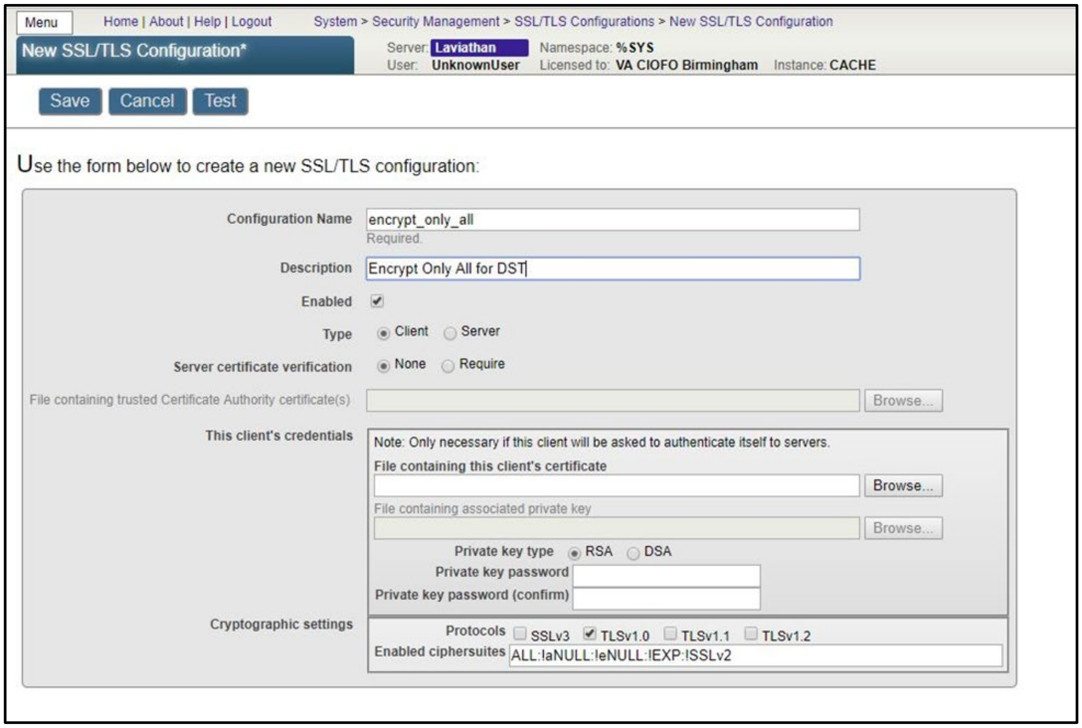
In order for the VistA component of DST (Patch GMRC*3.0*124) to connect to the DST servers, it is necessary to add SSL/TLS encryption to VistA using the Cache Management Portal. This MUST be done for ALL VistA systems whose CPRS Users need to use DST. The following steps describe how to do this.

1. Access the Cache Management Portal using either the appropriate URL for the site, or the link via the Cache Cube.
2. Navigate to the following: **System**, then **Security Management**, then **SSL/TLS**, and then **Configuration**.
3. Do one of the following:
    a. If the configuration “encrypt_only_all" already exists, then click **Edit** to view the configuration and verify that it is correct.
    b. If the configuration "encrypt_only_all" does NOT exist, then use the **Create New Configuration** option.
4. Verify that the following configuration options is set (as shown in Figure 4 and Figure 5).

o Configuration Name:	encrypt_only_all
o Description:	Encrypt Only All for DST
o Enable:	Checkbox should be checked.
o Type:	“Client” should be selected.
o Server Certificate Verification level:	“None” should be selected.
o File containing Trusted Certificate:	Leave blank.
o This client’s credentials:	Leave entire section blank.

5. Verify Cryptographic settings for Protocols:
    - For Cache 2014: TLSv1 should be checked. Uncheck the other boxes.
    - For Cache 2017: TLSv1.0 should be checked. Uncheck the other boxes.
6. Enabled ciphersuites should remain default as shown below:
    - Cache 2014 - TLSv1:SSLv3:!ADH:!LOW:!EXP:@STRENGTH
    - Cache 2017 - ALL:!aNULL:!eNULL:!EXP:!SSLv2
7. Click **Save** to save the configuration.

Figure 4: Creating a New SSL/TLS Configuration - Cache 2017


Menu
Home | About | Help | Logout
System > Security Management > SSL/TLS Configurations > New SSL/TLS Configuration
Save
Cancel
Test
Browse...

New SSL/TLS Configuration*

Server: Laviathan Namespace: %SYS
User: UnknownUser Licensed to: VA CIOFO Birmingham Instance: CACHE

Use the form below to create a new SSL/TLS configuration:

Configuration Name: encrypt_only_all
Required.

Description: Encrypt Only All for DST

Enabled: ☑

Type: ● Client ○ Server

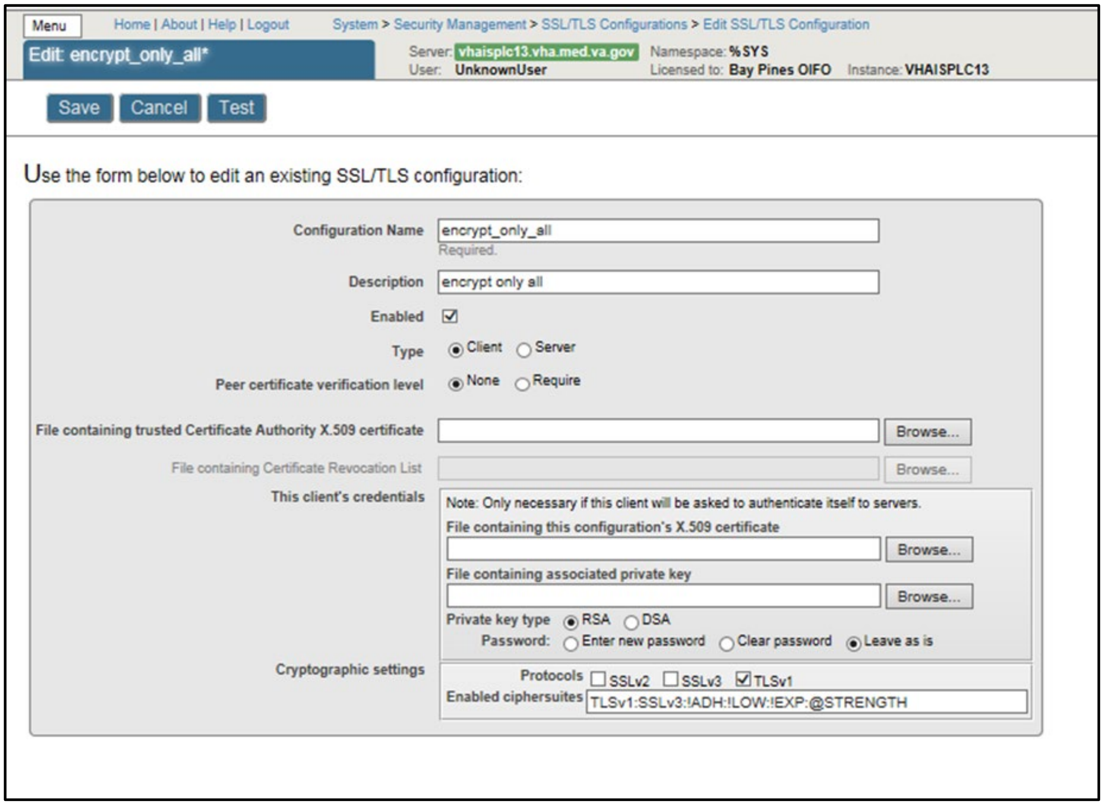
Server certificate verification: ● None ○ Require

File containing trusted Certificate Authority certificate(s)

This client's credentials: Note: Only necessary if this client will be asked to authenticate itself to servers.
File containing this client's certificate
File containing associated private key
Private key type ● RSA ○ DSA
Private key password
Private key password (confirm)

Cryptographic settings: Protocols □ SSLv3 ☑ TLSv1.0 □ TLSv1.1 □ TLSv1.2
Enabled ciphersuites ALL:!aNULL:!eNULL:!EXP:!SSLv2

Figure 5: Creating a New SSL/TLS Configuration - Cache 2014


Menu
Home | About | Help | Logout
System > Security Management > SSL/TLS Configurations > Edit SSL/TLS Configuration
Save
Cancel
Test
Browse...

Edit: encrypt_only_all*

Server: vhaisplc13.vha.med.va.gov Namespace: %SYS
User: UnknownUser Licensed to: Bay Pines OIFO Instance: VHAISPLC13

Use the form below to edit an existing SSL/TLS configuration:

Configuration Name: encrypt_only_all
Required.

Description: encrypt only all

Enabled: ☑

Type: ● Client ○ Server

Peer certificate verification level: ● None ○ Require

File containing trusted Certificate Authority X.509 certificate

File containing Certificate Revocation List

This client's credentials: Note: Only necessary if this client will be asked to authenticate itself to servers.
File containing this configuration's X.509 certificate
File containing associated private key
Private key type ● RSA ○ DSA
Password: ○ Enter new password ○ Clear password ● Leave as is

Cryptographic settings: Protocols □ SSLv2 □ SSLv3 ☑ TLSv1
Enabled ciphersuites TLSv1:SSLv3:!ADH:!LOW:!EXP:@STRENGTH

4.2.1 DST Application

Within the current CPRS order consult workflow, Veterans and care providers will utilize the DST system to support the decision and election for consult services for a given consult. The DST system will be integrated within the CPRS workflow to support all stakeholders. The DST web application is launched by the CTB application during the order consult workflow.

The DST web application displays all required decision data from the internal Department of Veterans Affairs (VA) data interfaces - Corporate Data Warehouse (CDW), Eligibility and Enrollment (E&E) Service, Provider Profile Management System (PPMS), Standardized Episodes of Care (SEOC), and Master Veteran Index (MVI), data entry functionality to capture a decision, and supportive data interfaces to save a decision to Veterans Health Information Systems and Technology Architecture (VistA) and Computerized Patient Record System (CPRS).

4.3 Download and Extract Files

DST does not download and extract files as a manual process. DST builds all environments using CI/CD pipeline approach utilizing a Jenkins build machine. With use of the Kubernetes infrastructure, as described in later sections, all services that comprise the DST application are compiled, packaged, published in the DST Jenkins environment. When a new deployment is available for an environment, the published Docker image artifacts are pulled from the registry to be installed within Kubernetes environment.

4.4 Database Creation

AWS RDS instance will be built with SQL script to contain the DST schema.

4.5 Installation Scripts

4.5.1 Install the Kubernetes Master and Worker

1. Log in to the Linux Master Kubernetes server and run the following commands.
    a. # sudo su -
    b. # cd ~
    c. # mkdir scripts
    d. # cd scripts
2. Copy the scripts onto the server and name it: **master-full-1-0-3.sh**.

```
#!/bin/bash
yum update -y
mkdir /root/custom_rpm_packages
# cd custom_rpm_packages/

# vi /etc/hosts

setenforce 0
sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
modprobe br_netfilter
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
swapoff -a
cat /etc/fstab

mkdir -p /etc/cni/net.d/
chmod -R 755 /etc/cni
cat <<EOF > /etc/cni/net.d/10-flannel.conf
```

```
 {
  "name": "cbr0",
  "type": "flannel",
  "delegate": {
    "isDefaultGateway": true
  }
}
EOF

chmod -R 755 /etc/cni

yum-config-manager --enable rhel-7-server-extras-rpms
yum-config-manager --enable rhui-REGION-rhel-server-extras
# /sbin/sysctl -w net.ipv4.conf.all.forwarding=1

yum install wget -y
wget -O /root/custom_rpm_packages/docker-18.06.rpm
https://download.docker.com/linux/centos/7/x86_64/stable/Packages/docker-ce-18.06.0.ce-
3.el7.x86_64.rpm
yum install http://mirror.centos.org/centos/7/extras/x86_64/Packages/pigz-2.3.3-
1.el7.centos.x86_64.rpm -y
yum install http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-selinux-
2.68-1.el7.noarch.rpm -y
yum install /root/custom_rpm_packages/docker-18.06.rpm -y

docker --version
systemctl enable docker && systemctl start docker
systemctl status docker

cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
        https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kube*
EOF

setenforce 0
yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

############################################################################################
##########################################################

systemctl enable kubelet && systemctl start kubelet
systemctl status kubelet
cat <<EOF >  /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sysctl --system
docker info | grep -i cgroup
/sbin/sysctl -w net.ipv4.conf.all.forwarding=1
sed -i 's/cgroup-driver=systemd/cgroup-driver=cgroupfs/g'
/etc/systemd/system/kubelet.service.d/10-kubeadm.conf
systemctl daemon-reload
systemctl restart kubelet

############################################################################################
##########################################################

systemctl disable firewalld
systemctl stop firewalld

kubeadm init --pod-network-cidr=10.244.0.0/16 > kubeadm_initial.log
KUBE_MASTER_IP=$(cat kubeadm_initial.log | grep "kubeadm join" | awk -F ' ' '{ print $3 }')
KUBE_MASTER_HASH=$(cat kubeadm_initial.log | grep "kubeadm join" | awk -F ' ' '{ print $7
}')
```

```
echo "If the master initialization was successful, then use the token bellow as it will
never expire."
kubeadm token create --ttl=0 > kubeadm_perm_token.log
KUBE_MASTER_TOKEN=$(cat kubeadm_perm_token.log)

# Make the needed env file
echo "" > worker-join.env
echo "export KUBE_MASTER_IP=$KUBE_MASTER_IP" >> worker-join.env
echo "export KUBE_MASTER_HASH=$KUBE_MASTER_HASH" >> worker-join.env
echo "export KUBE_MASTER_TOKEN=$KUBE_MASTER_TOKEN" >> worker-join.env

ec_user_home="/home/ec2-user"
mkdir -p $ec_user_home/.kube
cp -i /etc/kubernetes/admin.conf $ec_user_home/.kube/config
chown -R ec2-user:ec2-user $ec_user_home/.kube

###########################################################################################
##########################################################

kubectl --kubeconfig=/etc/kubernetes/admin.conf apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

###########################################################################################
##########################################################

echo "Copy the contents of this file and paste it into /root/scripts/worker-join.env on the
worker node, and then run the worker-full-1-0-3.sh script after."
cat /root/scripts/worker-join.env

sed -i --follow-symlinks 's/ipv6.disable=1/ipv6.disable=0/g' /etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Execute the shell script.
4. Restart the server.
5. Login to the Linux Worker Kubernetes server(s) and run the following commands.
    a. # sudo su -
    b. # cd ~
    c. # mkdir scripts
    d. # cd scripts
6. Copy the scripts onto the server and name it **worker-full-1-0-3.sh**.

```
#!/bin/bash
yum update -y
mkdir /root/custom_rpm_packages
# cd custom_rpm_packages/

# vi /etc/hosts

setenforce 0
sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g'
/etc/sysconfig/selinux
modprobe br_netfilter
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
swapoff -a
cat /etc/fstab

yum-config-manager --enable rhel-7-server-extras-rpms
yum-config-manager --enable rhui-REGION-rhel-server-extras
/sbin/sysctl -w net.ipv4.conf.all.forwarding=1
```

```
yum install wget -y
wget -O /root/custom_rpm_packages/docker-18.06.rpm
https://download.docker.com/linux/centos/7/x86_64/stable/Packages/docker-
ce-18.06.0.ce-3.el7.x86_64.rpm
yum install
http://mirror.centos.org/centos/7/extras/x86_64/Packages/pigz-2.3.3-
1.el7.centos.x86_64.rpm -y
yum install
http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-
selinux-2.68-1.el7.noarch.rpm -y
yum install /root/custom_rpm_packages/docker-18.06.rpm -y

docker --version
systemctl enable docker && systemctl start docker
systemctl status docker

cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
        https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kube*
EOF

setenforce 0
yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

systemctl disable firewalld
systemctl stop firewalld

#########################################################################
#########################################################################
###

systemctl enable kubelet && systemctl start kubelet
systemctl status kubelet
cat <<EOF >  /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sysctl --system
docker info | grep -i cgroup
/sbin/sysctl -w net.ipv4.conf.all.forwarding=1
sed -i 's/cgroup-driver=systemd/cgroup-driver=cgroupfs/g'
/etc/systemd/system/kubelet.service.d/10-kubeadm.conf
systemctl daemon-reload
systemctl restart kubelet

#########################################################################
#########################################################################
###
```

```
sed -i --follow-symlinks 's/ipv6.disable=1/ipv6.disable=0/g'
/etc/default/grub
grub2-mkconfig -o /boot/grub2/grub.cfg

source /root/scripts/worker-join.env

sudo kubeadm join $KUBE_MASTER_IP --token $KUBE_MASTER_TOKEN --discovery-
token-ca-cert-hash $KUBE_MASTER_HASH

#########################################################################
#########################################################################
###
```

8. Restart the server. When the servers restart, the installed Kubernetes environment will have a Master and worker node(s). To check this, run the following command:
   `#kubectl --kubeconfig=/etc/kubernetes/admin.conf get nodes`

   The running Master and Worker node(s) will be displayed.

4.5.2 Install the Apache Server

Please note the following pre-requisites prior to performing the steps below:

Pre-requisite: A valid internal VA certificate is installed on the Apache servers with the correct permissions for the Apache user. This certificate is used for communications with the SSOi SiteMinder agent that controls all authenticated traffic to the site. All certificates were registered on April 2019 and will expired in 2 years per VA policy. New certificates can be requested at the VA PKI site http://vaww.pki.va.gov/enrollment/index.asp.

Pre-requisite: A valid URL has been coordinated with the VA AD team and the proper A host record has been entered into the VA DNS to resolve to the Apache server.

1. On the Apache server, be sure the "httpd" process is started and set to automatically start at boot time.
2. Browse to the following directory:
   /etc/httpd/conf.d
3. Create a file named *dst.conf* and change the ownership of the file to APACHE.
4. Insert the following contents into the file:

NOTE: *This is an example file provided, the highlighted sections will have to change according to the environment being installed.*

```
<VirtualHost *:80>
        ServerName HOSTNAME.va.gov
        ServerAdmin DST-EMAIL-GROUP@va.gov
        UseCanonicalName on
        RewriteEngine On
        RewriteCond %{SERVER_PORT} !443
        RewriteRule (.*)$ https://dst.va.gov$1 [R]
</VirtualHost>

<VirtualHost *:443>
        ServerName HOSTNAME.va.gov:443
        ServerAdmin DST-EMAIL-GROUP @va.gov
        UseCanonicalName on
        LogLevel warn

        SSLEngine on
        SSLProtocol all -SSLv2 -SSLv3 -TLSv1
        SSLCipherSuite HIGH:!aNULL:+SHA1+MD5:+HIGH
        SSLCertificateFile /etc/httpd/certs/DST-CERTIFICATE.cer
        SSLCertificateKeyFile /etc/httpd/certs/DST-KEYFILE.key
        SSLCertificateChainFile /etc/httpd/certs/cacerts

#### Header Settings ####
        <IfModule mod_headers.c>
                Header always edit Set-Cookie
"(?i)^((?:(?!;\s?HttpOnly).)+)$" "$1; HttpOnly"
                Header always edit Set-Cookie
"(?i)^((?:(?!;\s?secure).)+)$" "$1; secure"
                Header always append X-Frame-Options SAMEORIGIN
                Header always set Strict-Transport-Security "max-
age=31536000; includeSubDomains"
                Header always set X-Content-Type-Options nosniff
                Header set X-XSS-Protection: "1; mode=block"
                Header always set Public-Key-Pins "pin-
sha256=\"base64+primary==\"; pin-sha256=\"base64+backup==\"; max-
age=5184000; includeSubDomains"
                Header unset Content-Security-Policy
                Header unset X-Content-Security-Policy
                Header unset X-WebKit-CSP
                Header add X-WebKit-CSP "default-src 'self'"
                Header unset X-Powered-By
                Header always set Access-Control-Allow-Origin "*"
                Header always set Access-Control-Allow-Headers "*"
                Header always set Access-Control-Allow-Credentials "true"
                Header unset ETag
                Header set Cache-Control "max-age=0, no-cache, no-store,
must-revalidate"
                Header set Pragma "no-cache"
                Header set Expires "Wed, 11 Jan 1984 05:00:00 GMT"
        </IfModule>

        ProxyPreserveHost On
        ProxyPass / http://Kubernetes-Worker:PORT/ connectiontimeout=900
timeout=900
        ProxyPassReverse / http://Kubernetes-Worker:PORT/

</VirtualHost>
```

5. Restart the HTTPD process.

4.5.3 Install SSOi on Apache Server

The SSOi installation is coordinated with the VA IAM team. Refer to the IAM teams page for the installation instructions here: https://vaww.oed.portal.va.gov/sites/vrm/IAM/playbooks/Pages/SSOi/CA%20SiteMinder%20WebAgent.aspx

4.6 Cron Scripts

No Cron Scripts are required for the DST application.

4.7 Access Requirements and Skills Needed for the Installation

Installers will need to have a proper ePAS in order to gain access to the server with elevated privileges. The installers will need to have knowledge of Apache, Kuernetes, Docker and GIT.

4.8 Installation Procedure

The installation software for DST is downloaded from the VA's GIT repo.

1. Clone/Pull GIT repo onto the Kubernetes MASTER server in the ~/deployments folder.
2. Browse to the dst-cprs-web folder:
   #cd ~/deployments/dst-cprs-web-release-vaec/charts/dst-cprs-web/
3. If there is an older release installed on the server previously, use this to remove the deployment:
   #helm delete --kubeconfig=/etc/kubernetes/admin.conf --purge dst-web
4. To install the latest release, use this command for the deployment:
   #helm --name dst-web --kubeconfig=/etc/kubernetes/admin.conf install.

4.9 Installation Verification Procedure

To verify the installation is running, use the following commands on the Kuernetes Master server:

1. *kubectl --kubeconfig=/etc/kubernetes/admin.conf get po*
   This will show all running pods on the server.
2. curl WORKERIP:WORKERPORT
   This will curl the front webpage to be sure there is a response.

4.10 System Configuration

This section is not applicable to the DST project.

4.11 Database Tuning

This section is not applicable to the DST project.

5. Back-Out Procedure

The steps described below outline the procedure to remove the DST application from the CPRS Platform in Production.

5.1 Back-Out procedure for the first deployment

1. Browse to the dst-cprs-web folder:
   #cd ~/deployments/dst-cprs-web-release-vaec/charts/dst-cprs-web/
2. If there is an older release installed on the server previously, use this to remove the deployment:
   #helm delete --kubeconfig=/etc/kubernetes/admin.conf --purge dst-web
3. Adjust the “tag” values in the values.yaml file to the installation versions to be installed.
   #vi values.yaml
4. Save the file.
5. To install the software, use this command for the deployment, this will read the values.yaml file:
   #helm --name dst-web --kubeconfig=/etc/kubernetes/admin.conf install .

5.2 Authority for Back-Out

Based on authority provided by our Business Sponsor and VA OIT IT program manager, DST can be backed out in accordance to their approval.

6. Rollback Procedure

1. Browse to the dst-cprs-web folder:
   #cd ~/deployments/dst-cprs-web-release-vaec/charts/dst-cprs-web/
2. If there is an older release installed on the server previously, use this to remove the deployment:
   #helm delete --kubeconfig=/etc/kubernetes/admin.conf --purge dst-web
3. Adjust the “tag” values in the values.yaml file to the installation versions to be installed.
   #vi values.yaml
4. Save the file.
5. To install the software, use this command for the deployment, this will read the values.yaml file:
   #helm --name dst-web --kubeconfig=/etc/kubernetes/admin.conf install.

6.1 Rollback Considerations.

DST can roll back any service within the Kubernetes cluster, which are all application components. DST can roll back the DST AWS RDS Postgres instance.

6.2 Rollback Criteria

Rollback criteria are not applicable.

6.3 Rollback Risks

There is minimal risk associated to these rollback procedures. It is common practice to rollback Kubernetes microservice and is part of the design of the technology. All DST application code and infrastructure are maintained as code that is saved in source control in VA GitHub So, there is minimal potential loss of functionality when an issue arises. Finally, AWS provides highly resilient backup processes for all AWS RDS databases.

6.4 Authority for Rollback

Based on authority provided by our Business Sponsor and VA OIT IT program manager, DST can be backed out in accordance to their approval.

6.5 Rollback Procedure

A rollback procedure is not applicable.

7. Risk and Mitigation Plan

The DST project team maintains a Program Risk Registry. Refer to the Program Risk Registry for all risks and mitigation plans for the entire DST project, including Consult Toolbox and VistA integration along with the rest of the VA partner interfaces (MVI, E&E, PPMS, CDW, SEOC).