JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

11100 Johns Hopkins Road • Laurel, Maryland 20723-6099

AOS-18-1655

5 November 2018

NIJ PROJECT 15-7

DFORC2 EVALUATION

Final Report

Version 1.0

Prepared for:
National Institute of Justice

NIJ | National Institute of Justice
STRENGTHEN SCIENCE. ADVANCE JUSTICE.

Prepared by:
The Johns Hopkins University
Applied Physics Laboratory
11100 Johns Hopkins Rd.
Laurel, MD 20723-6099

Task No.:	FGSGJ
Contract No.:	2013-MU-CX-K111/115912

This project was supported by Award No. 2013-MU-CX-K111, awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication/program/exhibition are those of the author(s) and do not necessarily reflect those of the Department of Justice.

This report provides a summary of the research and tasks completed by the RT&E Center for evaluation of the Digital Forensics Compute Cluster (DFORC2) tool created by the RAND Corporation. The DFORC2 tool seeks to increase throughput of digital evidence processing by performing computations in the Amazon Web Services cloud. However, at this stage, the tool is not ready for wide distribution and use, as its prototype status prohibited a functional installation of DFORC2 in a testing environment. DFORC2 and a similar tool from AccessData, AD Lab, were evaluated to the extent possible against a number of requirements developed by the RT&E Center for this task. In order to be adopted by practitioners, DFORC2 should be updated to operated with the current versions of its dependant software and services, and the installation process should be streamlined with complete documentation to allow smooth installation without the need for AWS and Kubernetes expertise. In this report, we present findings and recommendations for the DFORC2 tool and the processing of digital evidence in a cloud environment. This work was completed under NIJ Cooperative Agreement, Award No. 2013-MU-CX-K111/115912.

CONTENTS

1. INTRODUCTION

1.1 Purpose

As the provider for the National Institute of Justice's (NIJ's) National Criminal Justice Technology Research, Test, and Evaluation Center (RT&E Center), the Johns Hopkins University Applied Physics Laboratory (JHU/APL) was tasked to conduct independent evaluation of the Digital Forensics Compute Cluster (DFORC2) software developed by the RAND corporation, with the purpose of determining its applicability to the needs of law enforcement.

Our evaluation first focused on the development of requirements for the DFORC2 tool through background research and interviews with practitioners. The results of this phase of our study are outlined in Section 3. We then explored other state-of-the-art tools for comparison to DFORC2; a description of each is given in Section 4. DFORC2 and one of these tools, AccessData Lab (AD Lab), were evaluated against the requirements developed during the initial phase. The summary of our findings for each requirement is provided in detail in Section 5. Section 6 gives recommendations for the future development of DFORC2 and Section 7 presents conclusions based on the findings of the evaluation.

This final report provides a summary of these tasks and other activities performed as part of this evaluation effort, and provides recommendations for the development of the DFORC2 software and digital evidence processing in the cloud. This work was completed under NIJ Cooperative Agreement, Award No. 2013-MU-CX-K111/115912.

To summarize our findings: the inventors of DFORC2 did identify and target a legitimate bottleneck in the activities of investigators. However, the DFORC2 tool is not yet ready for general use. It is still very much a prototype and requires significant effort and troubleshooting to complete a working installation—a task that the RT&E Center does not recommend investigators undertake without substantial technical support. Though RAND reports noteworthy increases in performance through the use of the DFORC2 tool [1], due to numerous issues in the software and documentation provided, as described in Section 5.2.1, the RT&E Center was unable to achieve a working installation of DFORC2 after months of effort and interaction with DFORC2 developers at RAND. Other organizations are also developing tools for processing digital evidence in a distributed environment. AD Lab from AccessData, for example, provides a similar service to that of DFORC2. The RT&E Center completed rudimentary testing of the AD Lab tool to measure throughput, a summary of which is provided in Section 5.7.

Cloud providers such as Amazon Web Services (AWS) offer security measures for varying levels of critical data. Organizations should review the policies referenced in Section 5.1.1 of this report to ensure that the correct infrastructures and standards are maintained before contracting with a cloud provider. Cloud service providers charge a monthly fee for the use of their services. The fees vary depending on the amount of processing completed by the organization in a given month. Affordability is a concern in any installation, whether on-premises or off; an overview of the unique considerations in this area for cloud processing is provided in Section 5.4. To begin, the following subsection discusses the scope of this work and Section 2 provides background on the field of digital forensics processing and the DFORC2 tool.

1.2 Scope

Throughout the planning of tasks for this project, the assumption was made that the DFORC2 tool is functional and at a stage of development where it is ready to be evaluated by a third party. During our attempts to install the tool in our AWS environment, it became clear that the DFORC2 tool as delivered is still a prototype; the RT&E Center was unable to install a complete working version of the tool in its target environment. A functional installation of DFORC2 in AWS would require additional development which was out of scope for this evaluation. This report details the obstacles to adoption of the tool that we uncovered during our evaluation, recommendations for the improvement of the DFORC2 tool, and general recommendations for digital evidence processing in the cloud.

Our evaluation includes a basic evaluation of the AD Lab tool from AccessData. DFORC2 builds on the existing open source digital forensics tool Autopsy (a subject which is explained further in Section 2.2), and AD Lab builds on the Forensic Toolkit (FTK), another well-known tool in the field. In the limited functional evaluations we conducted, this evaluation focuses on throughput only. That is, we are not striving to compare the accuracy of the underlying tools; we are focusing only on speed and throughput.

2. BACKGROUND

2.1 Digital Forensics Tools

The use of digital forensics has grown tremendously during the past few decades, as an increasing amount of evidence is collected from digital devices. Despite a number of commercial and open source tools designed for digital forensics, it is interesting to note that many of these tools follow a similar evolution. In particular, most tools are evidence-oriented insofar as they assist investigators in finding evidence, not necessarily assisting with a systematic investigation. Quoting Garfinkel [2],

> Put crudely, today's tools were created for solving child pornography cases, not computer hacking cases. They were created for finding evidence where the possession of evidence is the crime itself.

Consequently, existing tools have a common conceptual model for investigations. In the "visibility" phase, data and metadata about digital objects (e.g., pictures and documents) are extracted and presented to the investigator. In the "filter" phase, the investigator can ignore irrelevant information such as details about operating system files. Finally, the "report" phase describes what was found and the process used to find it.

Despite its prevalence, the prior model tends to constrain the types of processing performed during a digital investigation. For example, an investigator may search for a specific credit card number, but triaging an investigation based on the set of likely credit card numbers found on a device (i.e., bulk data analysis [3]) is much less common. Furthermore, the visibility-filter-report model is not amenable to automation or parallel processing, leading to increasing delays due to a backlog of cases and increasing amounts of digital evidence being collected by law enforcement agencies.

2.2 The Digital Forensics Compute Cluster (DFORC2)

Under the auspices of NIJ, RAND Corporation developed the Digital Forensics Compute Cluster (DFORC2) [1] to provide a cost-effective and efficient digital forensics capability. An extension of the open source Autopsy digital forensics platform [4], DFORC2 parallelizes data ingest and file analysis for the processing of digital evidence found on suspect devices. DFORC2's distributed design allows it to scale efficiently when additional compute resources are available (e.g., a local cluster or third-party cloud[1]), reportedly improving performance by up to an order of magnitude for multi-terabyte disks [1].

The need for distributed processing by digital forensics tools has long been recognized [5] [6]. In a nutshell, disk capacity (commonly measured in GBs or TBs) has increased exponentially during the past few decades whereas other aspects of performance—such as the number of operations per second on those disks—have increased only linearly during the same time. Consequently, digital forensics investigators must sift through an increasing volume of data with minimal improvements to their tools. This asymmetry means that critical cases, such as kidnapping or terrorist threats, may remain open far longer than necessary. DFORC2 aims to remedy the existing gap through parallel data processing.

In DFORC2, data ingest is handled by dc3dd [7], an open source forensic tool that hashes and verifies data blocks. Data blocks are distributed via Apache Kafka [8], an open source stream processing platform designed to handle real-time data feeds. Apache Spark [9] provides a distributed framework for processing each data block, identifying files, storing metadata about those files, and reconstructing the original logical file system. Separate worker nodes perform digital forensics tasks, including hashing files (for comparison with databases of known files such as the National Software Reference Library (NSRL) [10]) and indexing strings to facilitate keyword searches by investigators.

To support its distributed data processing, DFORC2 requires a scalable, distributed storage system. The current implementation of DFORC2 targets Amazon Elastic File System (EFS) and Amazon Elastic Block Store (EBS), offerings that are part of Amazon Web Services (AWS) Elastic Compute Cloud (EC2). EFS, in particular, is an attractive option for digital forensics investigations because the storage size scales automatically, eliminating up-front costs and minimizing ongoing storage costs.

[1] A *cloud* uses virtualization to provide computing, storage, and even applications as a service across a network [29]. In many cases, the computing infrastructure is owned and managed by another organization, an arrangement that increases efficiency and decreases costs as compared to a traditional, on-premises data center.

Table 1 provides previously-published results comparing the performance of stand-alone Autopsy with DFORC2. As evidenced by the table, DFORC2 provides a substantial reduction in the length of time required to process data—approximately a three-fold improvement for a 232 GB disk image. In fact, the data ingest step dominates this time because dc3dd uses a single process for computing block hashes. However, processing these blocks essentially occurs in real-time. That is, once the data is ingested, minimal time is spent processing it. In comparison, stand-alone Autopsy spends the majority of time processing the data. For this evaluation, stand-alone Autopsy was run on a c3.2xlarge AWS instance (28 EC2 compute units (ECUs) and 15 GB RAM [11]). DFORC2 used 20-66 processing nodes, each a m4.large instance (6.5 ECUs and 8 GB RAM [12]).

Table 1: Performance comparison of stand-alone Autopsy and DFORC2 (from [1])

	Autopsy			DFORC2		
Image Size (GBs)	**Ingest (m)**	**Process (m)**	**Total (m)**	**Ingest (m)**	**Process (m)**	**Total (m)**
40	22.0	27.6	49.6	22.0	16.0	22.0
75	42.0	365.4	407.4	33.0	28.0	33.0
232	142.0	812.4	954.4	296.0	176.0	296.0

3. DEVELOPMENT OF REQUIREMENTS

The documentation provided by the RAND Corporation for DFORC2 unfortunately did not include formal requirements for the system. As such, supported by previous research in the field and the RT&E Center's software engineering expertise, a portion of the initial discovery efforts of this evaluation project were aimed at developing a set of functional and non-functional requirements for DFORC2. While this list was compiled specifically for the DFORC2 use case, the majority of these requirements may be applied to any digital forensics tool. The tasks we completed in this project were focused on the evaluation of DFORC2 in regard to the following requirements.

Data Integrity and Authentication
Digital information must not be modified without authorization, and evidence presented must support the finding that its proponent claims. In other words, the investigator needs to be able to prove that the digital evidence found on a suspect's device belongs to that suspect and has not been modified [13, 14]. Proper chain of custody should be maintained; if the data is modified during the investigation, the changes must be identifiable and must correlate to the process used.

Packaging
During our interviews with practitioners, we found that many law enforcement digital forensics labs do not have dedicated Information Technology (IT) staff. In order for tools to be used by the law enforcement community, they should be packaged in a way that promotes easy installation. The package installation should include or automatically install any dependencies, and the user should not have to manually compile code in order to install the product.

Usability and Support

This requirement encompasses ease of use and availability of help desk support and training for the tool.

Flexibility

For widespread adoption, DFORC2 should be capable of installation on both local infrastructure and remote infrastructure, such as AWS.

Affordability

The tool must be affordable for law enforcement agencies.

High-quality Source Code

This requirement comprises actions taken by the development team to ensure that their tool is dependable and does not include or introduce vulnerabilities to the system upon installation. To produce high-quality source code, rigorous software engineering practices should be maintained from the beginning of the project. More specifically,

- the development team should establish and adhere to a formal software development process for collaboration and communication;
- requirements should map to the system specification, which should then map to developed code;
- software should maintain supply chain integrity;
- source code should be subjected to unit, integration, and system testing throughout development, and this testing should include negative test cases;
- the tool should have proper use of third party libraries and code;
- source code complexity should be minimized;
- the system should be analyzed with both static and dynamic analysis tools, and any findings should be addressed; and
- production code should include no extraneous code.

Accuracy

This requirement is the need for a digital forensics tool to have reproducible results coupled with low false positive and false negative rates. The National Institute of Standards and Technology (NIST) has devoted resources towards this same goal in the form of the Computer Forensics Tool Testing (CFTT) Project [15]. The CFTT Project website hosts test procedures and data sets that tool makers can use to test digital forensics tools. Results of these tests should be published and reproducible so that all involved in the proceedings of a given case can make informed decisions in regard to the accuracy of the findings produced by a specific tool.

Processing Speed

The primary goal of DFORC2 is to decrease the amount of time it takes to process a given set of digital evidence. The performance improvement when compared to a representative single-user workstation then becomes an additional requirement of DFORC2 to be evaluated as part of this project. In addition, this performance improvement should alleviate an existing bottleneck in investigations that cannot be addressed through alternative approaches.

4. TOOLS FOR EVALUATION

During our research phase, in order to compare DFORC2 to the state-of-the-art in the field of digital forensics processing, we sought out additional tools for evaluation. AccessData Lab (AD Lab) from AccessData was quickly identified as a tool for comparison. We also investigated Turbinia and EnCase Forensic, but did not move forward with an evaluation of these tools. This section gives an outline of each tool and their relevance to this task.

4.1 DFORC2

The DFORC2 tool can be thought of as a distributed processing application that leverages Autopsy and The Sleuth Kit, but also has many other components as noted in [1]. Autopsy, detailed further in Section 4.2, has been modified for DFORC2 and uses a special *Image File to Cluster* module to start the ingest process. The modified version of Autopsy then communicates to containers running in the cloud environment to perform the processing. Theoretically, the frontend could be running remotely and communicate with the AWS cluster over the Internet via an elastic IP; however, network settings may restrict the ability for such a remote setup. The following are some key software components of DFORC2:

Dc3dd
The dc3dd tool is a well-known tool used to create, read, and manipulate disk images [7]. This tool is used by the modified version of Autopsy to send blocks to the Kafka Messaging Service. This occurs on the frontend of the DFORC2 application.

Kafka
Kafka is a distributed messaging service that is used by DFORC2 to take data published by dc3dd and send it to Kafka brokers that each handle a Kafka partition. Blocks from the partitions are then processed by Spark nodes. Thus, Kafka is necessary for the distribution of data from a single target image file.

Spark
These containers are managed by Kubernetes and pull blocks from the Kafka partition. They identify files in the block and store files as well as file system information in the Postgres database.

Worker Nodes
These containers, referred to as cluster worker nodes, run a completely separate modified headless version of Autopsy that handles tasks such as file hashing. The files identified by the Spark nodes can be further processed by these nodes. These nodes must be removed and restarted between cases. A script is provided by DFORC2 to automate this process.

Postgres
An instance of a Postgres database is run as a container on the Kubernetes cluster to store metadata information. This is used by the Spark nodes to store information on identified files. That data is then used by worker nodes to perform file level operations such as hashing or exif extraction.

NFS
This container is used to host the shared network file system (NFS), which is accessed by the Spark and Worker nodes.

Kubernetes
As stated in [1], Kubernetes is a key component of DFORC2. It is used to orchestrate the containerized applications upon which the backend of DFORC2 depends. These containers include but are not limited to Postgres, Spark, Kafka brokers, a NFS server, Apache solr[2] service, ActiveMQ[3] service, and Apache ZooKeeper[4] nodes. Following the documentation provided, a Kubernetes cluster is configured with at least one master and several worker nodes in separate instance groups. These instance groups correspond to virtual machine auto scaling groups that are instantiated in AWS. Once the initial set of virtual machines (VMs) has been created, the containers can then be deployed. Scripts were provided to help automate standing up the containers. This architecture separates the frontend of the application, which runs Autopsy, from the backend, which uses multiple containers and VMs to actually process the data.

To setup a DFORC2 cluster, the aws.conf file must be manually edited to include the correct AWS information such as the zone, S3 state store name, and network information. Then a script can be used to deploy the cluster.

4.2 Autopsy

Autopsy [4] is a digital forensics platform that provides a graphical user interface (GUI) to The Sleuth Kit (TSK) and other digital forensics tools. It is an open source tool and provides an interface for the incorporation of modules created by developers to provide additional functionality. Although Autopsy was not built with distributed processing in mind, as mentioned in Section 2.2, the DFORC2 tool uses Autopsy and TSK in the cloud to perform operations on digital evidence. As such, the Autopsy platform provides a good baseline for comparison of code quality to DFORC2.

The current version of Autopsy is 4.9.0, released on October 15, 2018. According to the RAND developers, the DFORC2 codebase branched off of the Autopsy ‘develop’ branch in GitHub at commit `#1ffeff74b54835384fd25f8c8dfa53d57a1839b5`. The closest stable build of Autopsy to that

[2] Apache solr is an open source search platform [31], used in DFORC2 for indexing
[3] ActiveMQ is an open source message broker from Apache [32]
[4] Apache ZooKeeper is “a central service for managing configuration information, naming, providing distributed synchronization, and providing group services”[30] in distributed applications.

commit is version 4.5.0. As such, our static analysis evaluation included Autopsy v4.5.0 and Autopsy v4.9.0.

4.3 AD Lab and the Forensic Toolkit (FTK)

The Forensic Toolkit (FTK) from AccessData (AD) is a well-known tool in the field of digital forensics. Product descriptions of FTK report the capability to handle up to four distributed processing engines (DPEs) in FTK alone [16]. AD Lab is another tool from AccessData, which includes FTK and enables distributed processing of digital evidence in the AWS Cloud. In a nutshell, AD Lab is to FTK what DFORC2 is to the Autopsy platform. The AD Lab platform can reportedly scale up to 16 DPEs and is also compatible with AWS Cloud [17].

We obtained an evaluation license for version 6.4.0 of AD Lab from AccessData to compare the tool’s distributed processing capabilities in AWS to that of the DFORC2 tool. AD Lab was installed in our AWS instance using a CloudLaunch template provided by AccessData for installation. The specific details of our installation used for testing are described further in Section 5.7.1.

The AD Lab cluster in AWS comprises four instances and an auto scaling group of variable size. All of the instances are Windows machines and use Active Directory for user login. The four main instances are:

BastionHost
This node has an external IP address which enables the user to make a remote desktop connection to the machine. All other instances are housed on the AWS Virtual Private Cloud (VPC) for AD Lab, and can be reached from the BastionHost through a remote desktop connection. The BastionHost can also be reached via ssh with a public/private key pair created during cluster deployment.

AD-MSSQL
This node hosts the Microsoft SQL (MS SQL) database server for the cluster. During cluster setup, the user is given the option either to use a SQL license from AWS or to bring their own license and use it in the cloud. We chose the former option in our setup.

DPM
The Distributed Processing Management (DPM) Server. This node splits processing tasks between instances in the auto scaling group. It also hosts the Evidence and Cases drives, which are shared across nodes. The Evidence drive is where evidence should be placed in order to be processed.

LabClient
This instance is the primary one an investigator would interact with after creating a remote desktop connection through the BastionHost. This node runs the interface for the AD Lab client software and the license manager for the tool. From this machine, the user can direct the DPM to process evidence found in the shared evidence directory and can view the results in the FTK GUI.

The final type of instance that is used in AD Lab is the DPEGroup (Distributed Processing Engine Group) instance. These instances are managed by the auto scaling group for the cluster. Each acts as a resource that the DPM can take advantage of for evidence processing. The number of DPEGroup instances is managed through the AWS console, by editing the desired number of nodes. It should be noted that this is not yet a fully-functional auto scaling group. In a complete auto scaling group, the number of nodes would be managed automatically by AWS based on processing load. In AD Lab, the investigator must manually set the number of nodes they desire. Then, the investigator must log into each individual node and start the Distributed Processing Engine (DPE) service on that node. This process could be scripted, but is not at the time of this writing.

All DPEGroup instances can be terminated by setting the desired number of nodes in the auto scaling group to zero (0). The other four main instances (BastionHost, AD-MSSQL, DPM, and LabClient) can be stopped from the AWS console without any negative effects. This is beneficial to users who wish to be more conservative with AWS usage. AWS will only bill users for the time the instances are up and running.

Depending on network security measures in place at the investigative site, remote desktop protocol (RDP) from within the enterprise network to an AWS instance may not be permitted. Permissions for RDP should be reviewed with the investigator's network administrators before usage of AD Lab.

In order to maintain a working AD Lab cluster, Windows Firewall had to be turned off on each of the AD Lab instances. Security akin to a firewall is maintained through the AWS security groups created during cluster instantiation. The rules for this security group are managed through the AWS console, and RDP can be limited to a range of IP addresses, or even individual IP addresses within an organization. RDP requires a password, and each user should have a unique login. Multi-factor authentication (MFA) was not enabled for our installation for RDP; however it is supported by Microsoft Remote Desktop Services and could be employed.

To process a case with AD Lab, the investigator follows these steps:

1. Specify the number of nodes to have in the auto scaling group through the AWS console interface.
    a. If any change was made to the number of instances or their IP addresses, the investigator must login to the DPM (by hopping through the BastionHost) to update the IP addresses of the DPEGroup instances used.
    b. If any new DPEGroup instances were brought up, the investigator must also log in to each of these to start the Distributed Processing Engine Service.

2. Load the evidence onto the DPM. This can be done a number of ways; the easiest we found (though not the most efficient) was to upload the evidence to an Amazon S3 bucket and download it onto the DPM through the AWS console. Alternatively, if the evidence is loaded onto a separate EBS volume, the volume can be attached to the DPM and shared

with the other instances to allow processing to complete, without the need to download the evidence.

3. Once the DPEGroup instances are ready and the evidence is in place, the investigator logs into the LabClient (again by hopping through the BastionHost) and starts the AD Lab software. They then select the location of the evidence and begin processing on the DPM through the FTK interface. The results populate the FTK GUI and can be explored there.

4.4 Other Relevant Tools

4.4.1 Turbinia

Turbinia is a Google-owned framework for "deploying, managing, and running forensic workloads on cloud platforms" [18]. Our analysis examined the master branch of their development at commit `#ca58f585bf6c613c8682bfa7b2f6ec493f86db1f`. Turbinia operates on a job-based system; each piece of evidence is combined with the selection of individual forensic tasks capable of processing the evidence data. Turbinia is able to process jobs and store evidence both locally and remotely, though it relies on Google Cloud PubSub Task Queue in both modes.

Turbinia first drew our interest because, similar to DFORC2, it uses The Sleuth Kit (TSK) on its backend. However, the scalability offered by Turbinia applies to scaling the number of jobs, allowing for high volume of discrete forensic jobs to be run simultaneously. It does not appear to support distribution of an individual job for higher throughput and, as such, does not provide a relevant comparison for performance to tools which support this functionality. In addition, at the time of the development of our test plan, Turbinia was described by its developers as 'pre-Alpha,' and still had many milestones to meet before the release of their initial 'Alpha' version. As such, the development required to make the Turbinia tool functional and a relevant comparison to DFORC2 was deemed out-of-scope for this evaluation.

The 'Alpha' version of the tool was released shortly before the writing of this document, and is a potential tool to consider for future research.

4.4.2 EnCase Forensic

EnCase Forensic is a digital forensics tool from OpenText (formerly Guidance Software) widely used by law enforcement to recover evidence from suspects' hard drives. Various publications from the developers of EnCase note the ability to distribute digital forensics processing across EnCase Processor Nodes. In this setup, each node requires a unique software license and security key from the vendor. Once configured, the EnCase Processor Server can distribute tasks across all registered nodes. We reached out to OpenText to inquire about the possibility of evaluating EnCase as part of this task, but due to lack of communication from OpenText, were not able to acquire the necessary licenses to include the tool in this evaluation. From the publications we've reviewed, it appears that the distributed processing capability relies on a local cluster and has not yet been adapted to operate in a public cloud such as AWS. [19]

5. EVALUATION OF REQUIREMENTS

As outlined in the Project Plan, the RT&E Center sought to evaluate DFORC2 and related tools against the requirements described in Section 3. Without a working installation of DFORC2 this evaluation is limited. Nonetheless, this section provides a summary of the tasks completed and findings under each requirement, as well as the results of testing the AD Lab tool in our AWS environment. Further technical details of the tasks completed are provided in the appendices and referenced in the appropriate sections. Section 6 gives a summary of the major recommendations enumerated in these subsections, both for the future of the DFORC2 tool and for investigators making decisions in regard to the adoption of tools such as these and cloud infrastructures.

5.1 Data Integrity and Authentication

Data integrity and authentication refers to the requirement that digital information must not be modified without authorization, and evidence presented must support the finding that its proponent claims. Investigators need to be able to prove that the evidence found on a device has not been modified since collection and that modifications done during processing are identifiable and attributable to the processes followed.

5.1.1 Cyber Security Impact Evaluation and Recommendations

As cloud infrastructures are not yet a common environment for digital evidence processing, investigators should carefully consider the impact of placing their digital evidence in these environments. There is no one-size-fits-all approach to managing cybersecurity risk for law enforcement organizations (LEOs) as each will have unique risks and vulnerabilities, face different threats and ultimately, have different risk tolerances based on their mission, budget, sensitivity of their data and impact of compromise or loss of information technology (IT) capabilities or data. Depending on the sensitivity of the data in question, different cloud vendors may be acceptable for its storage and processing. Therefore, LEOs should reference cybersecurity guidance and standards published by US Federal organizations and national and international standards bodies as well as industry to determine the necessary level of protection for their data, and the best approach and infrastructures to use to maintain the necessary level of security in both traditional and cloud infrastructures.

Appendix A – Security Guidance and Standards provides a detailed overview of the documents relevant to this evaluation.

5.1.2 Basic Security of Evaluated Tool Installations

As described in the previous section, LEOs should review the impact level of their data and determine the applicable guidance for cloud infrastructure provisioning. Even with the correct infrastructure in place; however, an insecure tool can still provide opportunities for exploit.

When deploying DFORC2 it is necessary to store AWS credentials (the security access key and access key id) on the machine from which the Kubernetes cluster is controlled. Through discussions with RAND and practical experience deploying the cluster, it was revealed that during development, this particular machine was also virtualized on AWS. Thus anyone with access to this VM can potentially impersonate the Kubernetes administrator. Although this VM is protected by a public/private key pair, if the permissions of the stored AWS credentials are too open, unnecessary risk could be taken on by a DFORC2 user.

An AWS AD Lab installation is accessed by RDP to the BastionHost instance. As noted in Section 4.3, access to the RDP interface is controlled through the AWS console by IP address and optional MFA. Each user of AD Lab should have their own license and authentication information, and good password security practices should be followed. Also as mentioned in previous sections, RDP access to the AWS machines could be limited by your network administrator. Organizations should discuss options with administrators before deciding on a specific tool/infrastructure.

When deploying a digital evidence processing capability on the AWS cloud, it is tempting to use S3 buckets to hold evidence, as they are one of the least costly options for storage in the cloud. The policies outlined in Appendix A should be consulted to determine whether this is appropriate for the impact level of your data, and common-sense measures should be taken such as using encryption and disallowing public access to data [20].

5.2 Packaging, Usability, and Support

In order to be usable by a wide variety of investigators with varying levels of technical expertise, a digital forensics tool should be straightforward to install and use, and should ideally offer support in the form of tutorials, video demonstrations, and/or helpdesk support. These requirements were not formally examined as part of this effort; however, the following section provides comments on the RT&E Center's experience in these areas for the evaluated tools.

5.2.1 DFORC2

DFORC2 was provided to the RT&E Center in two separate forms. The first was a collection of zip files that contained pre-built containers. The second was as a collection of GitLab repositories from which DFORC2 could be built from source. Deploying DFORC2 is challenging in its current state. It appears that the project has not been steadily maintained since its delivery to the sponsor, which has resulted in many software compatibility issues. There appears to be no formal documentation for deploying DFORC2 on AWS. The documentation is segmented in several README files contained in various repositories. The AWS-Kube-User guide that was provided is sparse, mainly containing links to other resources. This collection of documents may be sufficient for more technical users; however, users with less experience with open source software and cloud development may find it highly frustrating. After troubleshooting various issues in the resources provided, we deployed a DFORC2 cluster in AWS but were unable to complete a full working installation of the software itself.

Appendix B – DFORC2 Technical Obstacles provides the technical details of the primary issues encountered when attempting to install and run DFORC2.

5.2.2 AD Lab

AD Lab targeted a purely AWS-centered installation method. Section 5.3 will discuss the decreased flexibility in this approach. The AD Lab cluster is created by subscribing to the AD Lab product on the AWS Marketplace. The marketplace walks the installer through a couple installation steps, where the user can choose to use their own AWS CloudFormation template or install the one provided on the Marketplace. The template then drives the remainder of the installation, where the user can select different instance types and licensing options for every node in the AD Lab framework. After these selections, the cluster is created and launched within AWS.

After the cluster is created, the user must log into each machine and perform some setup tasks, including disabling the Windows firewall, changing passwords, sharing the evidence and case folders on the DPM, and finally registering license keys for AD Lab itself. The user must also manage configuration of Active Directory and Microsoft SQL Server, and these tasks can vary in difficulty depending on whether the organization is utilizing an existing instance of each of these, or creating new ones within AWS. The installation steps we followed were provided to us by AccessData, and can also be downloaded from the AD Lab product page as of this writing [17].

The AD Lab software from AccessData is still under development, and the installation process was not without obstacles. However, the RT&E Center independently deployed the AD Lab cluster successfully in less than a day, then was able to perform a processing task on a test image after troubleshooting the installation of the software itself with AccessData representatives for a total of four hours. This troubleshooting primarily involved a complete re-install of the MS SQL database, as well as re-configuring of the launch template for the autoscaling group. AccessData is aware of these issues and we did see an improvement of various aspects during our interactions with the team; that is to say, the RT&E Center expects the AD Lab installation process to improve.

After this process, we had a fully working installation of AD Lab and were able to process a test image. All of the AD Lab instances are Windows 10 machines, thus they will be most usable to those investigators comfortable working in a Windows environment. Experience with MS SQL and Active Directory is also beneficial for configuration and setup.

5.2.3 Auto Scaling Groups

Auto scaling groups in AWS allow the cluster to grow and shrink according to processing need. This allows a cluster to only maintain (and incur fees for) nodes that it actually needs for a processing task. Neither AD Lab nor DFORC2 fully support auto scaling groups in AWS. In both tools, the investigator needs to determine the ideal number of nodes for a given case (or use the same number of nodes for every case). AWS will then spin up new auto scaling group instances according to the number set in the AWS console. However, in each tool, the investigator must log in to each individual node and perform setup tasks manually, including restarting services on the node. For each tool, we recommend that this process be improved to allow true auto scaling capabilities, first by eliminating the need to log into each node after creation, and second, by fully utilizing AWS's capabilities to identify and automatically spin up more nodes as needed, instead of as determined ahead of time by the investigator. Until full auto scaling is implemented, the usability of the tools is slightly decreased as a certain level of technical expertise and time is required to perform these start up tasks on each node.

5.3 Flexibility

Ideally, a digital forensics tool should support all of the platform(s) desired by investigators. However, developing the tool for each environment and type of infrastructure is not an insignificant task. While Autopsy officially supports Windows, the developers also provide installation instructions for Mac OS X and Linux-based hosts. DFORC2 does not maintain the Autopsy support for Windows, and only operates on Linux instances in AWS. This automatically decreases the flexibility of DFORC2, as Autopsy does not support full capability outside of the Windows operating system. AD Lab only supports Windows installations in AWS.

As explained in Section 5.1.1, there are multiple cloud vendors with varying levels of acceptability for the processing of high-impact data. Both AD Lab and DFORC2 only support the AWS cloud as their installation infrastructure. As AWS is one of the few providers that maintains support for high-impact data, AWS is a primary candidate for deployment; however, support for other cloud infrastructures would improve investigator's abilities to maintain flexibility in their choice of tool/deployment scenarios.

As such, we recommend the developers of each tool investigate support of other cloud infrastructures. At the time of this writing, AccessData lists Microsoft Azure support as upcoming. It should be noted that AccessData also reports support for up to four Distributed Processing Engines (DPEs) through the use of FTK alone. This installation was not investigated by the RT&E Center team, but likely uses highly-capable on-premises infrastructure to distribute tasks across machines.

As part of our evaluation, we investigated the level of effort necessary to port DFORC2 capabilities to an OpenStack infrastructure as a means of determining the flexibility of the tool. Although not officially supported by the tool, publications from RAND cite the intention to apply DFORC2 to other infrastructures [1]. The RT&E Center discovered that the DFORC2 codebase is very AWS-specific, and porting the tool to OpenStack quickly became too resource-intensive to fit within the scope of this evaluation. The results of this investigation are summarized in Appendix C – DFORC2 on Openstack.

5.3.1 Uploading Evidence

Also related to flexibility, we recognized a level of decreased flexibility in our AWS installation of AD Lab due to the requirement of Windows machines for the tool. AWS does not yet officially support Elastic File Storage (EFS) mounting or S3 bucket application program interface (API) calls for Windows machines. As a result, moving evidence to the AD Lab instances requires workarounds. The AD Lab installation instructions enumerate a couple of methods to upload the evidence files, including creating a virtual private network (VPN) connection to the AD Lab virtual private cloud (VPC), or uploading the evidence to an S3 bucket then downloading it to the machine. The former requires further AWS configuration expertise, as well as coordination with the investigator's on-site IT services. Using the latter suggestion, the evidence is downloaded from the S3 bucket by logging into the AWS console from the on the DPM instance. This download will take a significant amount of time for large amounts of evidence. For the purposes of our testing, we created test images on an EBS volume attached to a Linux machine, detached the volume, and attached to the DPM for processing. This process is explained further in Section 5.7.

Although DFORC2 maintains support for S3 buckets and EFS volumes through the use of Linux machines for installation, uploading evidence to the storage volumes is identified as an obstacle by the RT&E Center for both tools.

5.4 Affordability

It should come as no surprise that different investigative organizations will have different funding concerns when considering a new digital evidence processing tool. Indeed, this was one of the main concerns that was raised by practitioners during interviews conducted as part of our research phase of this evaluation. Funding concerns are largely outside the scope of this evaluation; however, this section should provide an overview of the items to consider when estimating costs for different infrastructures and tools.

AWS incurs a monthly charge based on the resources used during that month. More capable instance types are more expensive than ones with fewer resources. In general, users only pay for the instances actually used during the time period. That is, if an instance is ‘stopped’, it no longer incurs a fee. Readers should consult AWS for the current pricing schedule. Table 2 gives an overview of the pricing schedule for various pertinent AWS EC2 instances at the time of this writing.

Table 2: Hourly cost of a selection of relevant EC2 on-demand instances in the US East - Virginia region as of November 2018 [12]

	Operating System	Instance Type	Cost per Hour
DFORC2 Master	RHEL	t2.medium	$0.1064
DFORC2 Node	RHEL	m4.xlarge	$0.2600
AD Lab Master (used)	Windows	c4.2xlarge	$0.7660
AD Lab Node (used)	Windows	m4.2xlarge	$0.7680
AD Lab Master (default)	Windows	c4.4xlarge	$1.5320
AD Lab Node (default)	Windows	i3.2xlarge	$0.9920

The total fees incurred by our evaluation are not enumerated here because there are various mitigating factors. First, we installed both the DFORC2 and AD Lab clusters with instances that were chosen from a set of possible types. Different selections during the installation process will incur different fees. Second, our consumption of the services was not the same as what we would expect from a normal investigative organization. After the initial setup of AD Lab we maintained the cluster in AWS, but ‘stopped’ for a long period of time during which it was only incurring storage fees—a lower cost than we would expect an investigative organization to incur. During the DFORC2 cluster deployment, the RT&E Center was repeatedly re-deploying instances and sometimes maintaining more nodes than necessary to troubleshoot issues, thus likely consuming more resources than we would expect from a normal investigative organization.

When facing the monthly fees of AWS, on-premises OpenStack cloud installations can begin to look more attractive. The one-time fee for equipment is sometimes a draw for organizations. However, this view is somewhat short-sighted. The mere configuration of an OpenStack cluster is no small feat, and will require regular maintenance throughout its lifetime. Hardware will eventually need to be replaced or augmented. In short, an on-premises cloud will not continue to operate smoothly indefinitely without dedicated IT support. Organizations that pursue this method should be aware of the resources required for upkeep. This is one of the main draws of AWS: resources are maintained and made available by Amazon.

In addition to the infrastructure costs, our installation of AD Lab in the AWS cloud included the cost of Microsoft SQL Server licenses from AWS. There is an option to bring your own license if your organization already has one available. AD Lab licenses are also needed for each user of the AD Lab software. Current pricing structures of this software should be discussed with AccessData representatives. AD Lab instances can be stopped when not in use, minimizing the total cost of the cluster. After stopping all instances, all of those outside of the AD Lab auto scaling group can be restarted though the AWS console. As of this writing, the DPEGroup instances will take longer to make operational after a shutdown, as the investigator must perform the manual steps referenced in Section 5.2.3.

As RAND's model for DFORC2 deployment is an open source release, the DFORC2 software does not come with an annual license cost. However, as open source software, users cannot expect dedicated support during the installation and/or troubleshooting process. The cost of the resources required to perform your own training, setup, troubleshooting, and maintenance should be considered. As noted in Section 5.2.1, our team was required to delve deep into the fundamentals of AWS and Kubernetes orchestration in order to make even a small amount of progress on the DFORC2 installation in AWS. The training and expertise required for this should not be neglected on cost estimates. Similar to AD Lab, DFORC2 instances can be stopped when not in use, but processing nodes will require manual startup activities after a re-deployment. With a higher number of nodes, this could become very cumbersome. The adoption of DFORC2 could be improved by allowing instances to be stopped when not in use and by improving the installation process to mitigate the need for cloud infrastructure expertise.

In addition to the expenses already enumerated, uploading and downloading data to/from the AWS cloud incurs additional fees. This varies, depending again upon the model an organization adopts. Upload of data to EC2 instances is free at the time of this writing. There is also the option to mail data to Amazon for upload to AWS instances at an additional fee [21]. Storage and download pricing depends on the amount of data to be stored and the frequency of access necessary [22, 23].

5.5 High-quality Source Code

As DFORC2 is the primary object of this evaluation and other evaluated tools are proprietary, this section focuses only on the aspects of the DFORC2 codebase that the RT&E Center could evaluate post-production. Without insight into the practices maintained during the development of the DFORC2 code, we cannot speculate on many of the desirable properties described herein.

As outlined in Section 3, high-quality source code should be developed while following rigorous software engineering practices. Namely,

- the development team should establish and adhere to a formal software development process for collaboration and communication;
- requirements should map to the system specification, which should then map to developed code;
- software should maintain supply chain integrity;
- source code should be subjected to unit, integration, and system testing throughout development, and this testing should include negative test cases;
- the tool should have proper use of third party libraries and code;
- source code complexity should be minimized;
- the system should be analyzed with both static and dynamic analysis tools, and any findings should be addressed; and
- production code should include no extraneous code

As previously mentioned, the DFORC2 project evaluated by the RT&E Center did not provide a set of formal requirements or specifications for the system. The DFORC2 code is currently housed within a private GitLab repository, which gives some indication of supply chain integrity in future development. In the repository, commits should be attributable to specific developers. The original Autopsy and TSK codebase are maintained in GitHub version control. The DFORC2 source code in GitLab is devoid of history, so that the RT&E Center cannot attest to the maintenance of integrity between the fork of the open source Autopsy and TSK repositories and the current state of the DFORC2 source code.

We performed static analysis on the DFORC2, Autopsy, and TSK codebases to give a basic understanding of the general quality of the DFORC2 source code compared to the original versions. The results of this task are summarized in the following section.

5.5.1 Static Analysis Results

In static code analysis, source code is examined for bugs without executing the actual code. The examination is performed against a set of defined coding rules that can be used to determine compliance with established and agreed-upon coding guidelines. In contrast, dynamic code analysis examines the code during execution. We performed static code analysis of the open source Autopsy and TSK source code and compared the results against RAND-modified versions of the same codebases for use in DFORC2.

We used Cppcheck version 1.84 for C/C++ analysis and SpotBugs version 3.1.3 with the find-sec-bugs security extension for Java code analysis [24, 25, 26]. As Autopsy is largely Java code and TSK is primarily C code, SpotBugs was applied to the former and Cppcheck to the latter. The RT&E Center did not find any significant differences between the open source versions of the code bases and the ones modified by RAND for use in DFORC2. Indeed, all bugs present in the modified versions are also present in the open source codebases. Cppcheck found 40 issues in TSK for both code bases, and SpotBugs with find-sec-bugs identified 145 issues in the Autopsy v4.9.0 codebase. Developers should continue static analysis tasks to identify and mitigate all bugs found.

This relates to another recommendation from the RT&E Center for the improvement of DFORC2. RAND has cited an intention to release the DFORC2 codebase as open source, but to our knowledge they have not outlined a plan to merge their changes back to the existing master repositories, or to apply changes from the master repositories to their code base. If open source developers find and fix a bug in Autopsy, with the current model, the DFORC2 code doesn't have an easy way to receive and apply this patch. This is a huge obstacle for future adoption of the tool, and could leave DFORC2 open to vulnerabilities long after appropriate patches are released. For reference, DFORC2 uses roughly Autopsy version 4.5.0 with RAND modifications. As of October 2018 Autopsy is at version 4.9.0, and DFORC2 has not incorporated any of the changes that took place between the two versions. The development of open source tools moves quickly; new versions that include big fixes and feature enhancements emerge daily. DFORC2 code is currently behind the curve in adoption of changes, a fact which will become more and more apparent as time progresses.

5.6 Accuracy

Accuracy is the requirement that digital forensics tools have reproducible results with low false positive and negative rates. These results should be public so that the community can compare the accuracy of tools and compare results across test sets. Since this requirement is already well-evaluated by organizations such as NIST [15] and our evaluation focused on throughput and the repercussions of moving digital forensics processing to the cloud, the requirement for accuracy was omitted from our evaluation. DFORC2 is based on TSK / Autopsy tools, open source digital forensics programs that are well-accepted by the community. AccessData's AD Lab tool uses FTK, which is also recognized as an acceptable tool in the digital forensics community.

5.7 Processing Speed

For our evaluation, evidence processing speed was a primary concern. DFORC2's main contribution is the distribution of evidence processing across nodes in AWS, in an effort to ease the bottleneck that exists in single-use workstations housed at investigative facilities. As described in Section 2.2, DFORC2 is derived from Autopsy, and the primary goal of DFORC2 is to decrease the amount of time it takes to process a given set of digital evidence. Interviews with practitioners confirmed that this is an area of concern in their investigations.

Unfortunately, the RT&E Center was unable to install an operational version DFORC2 in a distributed environment. RAND-reported results in processing images of varying sizes in AWS are provided in Table 1, and we completed an evaluation of the AD Lab tool from AccessData to demonstrate a comparable state-of-the-art tool in the field.

5.7.1 AD Lab – AWS Installation Details

Our AWS instance was provided by the JHU/APL Information Technology Services Department (ITSD). It is operating in the AWS us-east-1 region. Note that this area of AWS is not approved for high-impact data under FedRAMP (see Appendix A). We chose it for our testing due to its ease of acquisition and our lack of personally identifiable information or other sensitive data in our test sets. Investigators should consult the policies referenced in Section 5.1.1 to determine the best infrastructure for their processing.

For the AD Lab installation, we used a YAML[5] CloudFormation template provided to us by AccessData on May 21, 2018, labeled version 2.6.0 and the default values provided therein, again with the exception of various node types. The default Distributed Processing Engines (DPE) node setting uses a i3.2xlarge instance. As shown in Table 3, this instance is much more capable than the default used by DFORC2. In order to more accurately compare the two implementations and limit cost, the DPEs in our AD Lab installation were chosen to be m4.2xlarge instances. To minimize costs, we decided to use a c4.2xlarge instance type for the Distributed Processing Management (DPM) Server, instead of the default c4.4xlarge instance type. Both of these modifications are provided as options in the AD Lab template and should be supported as such. These choices definitely affected the overall performance of AD Lab. Along with housing more memory and more-capable virtual central processing units (vCPUs), more capable instance types also often have higher bandwidths on the AWS network. Variable instance types were not explored during our evaluation.

Table 3: Comparison of DFORC2 and AD Lab instances [43]. For clarity, the AD Lab-specific names DPM and DPE are replaced with the generic names for the equivalent functions: Master and Node.

	Instance Type	**vCPU**	**ECU**	**Memory (GB)**	**Instance Storage**
DFORC2 Master	t2.medium	2	Variable	4	EBS Only
AD Lab Master (used)	c4.2xlarge	8	31	15	EBS Only
AD Lab Master (default)	c4.4xlarge	16	62	30	EBS Only
DFORC2 Node	m4.xlarge	4	13	16	EBS Only
AD Lab Node (used)	m4.2xlarge	8	26	32	EBS Only
AD Lab Node (default)	i3.2xlarge	8	27	61	1 x 1900 NVMe SSD

The LabClient and BastionHost machines were t2.large instances, and the AD-MSSQL instance was an m4.xlarge instance. AD Lab was deployed with 1, 2, 4, 6, or 8 DPEs to test the ability of the tool to farm out processing to a different number of nodes at varying amounts of data. The exact tests run are described in the next section.

While AD Lab lists a number of ways to upload evidence to the DPM, there is no recommended method as of this writing [17]. For our testing purposes, the custom images described in Section 5.7.2 were created on an EBS volume on a Linux machine in our AWS cluster, then the EBS volume was detached from the Linux machine and attached to the DPM to be shared and processed as evidence. Creation time of the images was not tracked. The DigitalCorpora image referenced in Section 5.7.2 was downloaded directly from DigitalCorpora.org to the Evidence folder on the DPM.

[5] YAML is a data serialization language commonly used for configuration files

5.7.2 Test Images

For testing, we employed a data set from DigitalCorpora.org, as well as our own custom test images to zero in on the processing power of the tools evaluated. DigitalCorpora.org maintains a repository of disk images that are targeted towards computer forensics education research and can be used without Institutional Review Board (IRB) approval. However, the primary focus of many of the well-known test images is accuracy of findings. Since the main topic of this evaluation and the primary concern with the core set of features listed is simply throughput, we also created our own custom images to test processing and scaling capabilities for these tools when presented with larger images.

The resulting images were 100, 200, or 250 GB, using either the New Technology File System (NTFS) or Fourth Extended Filesystem (Ext4)[6]. Table 4 contains a summary of the images downloaded/created for testing, their file system type and size. A description of the process for creating the custom images follows. To create these images, we used the files provided in DigitalCorpora.org's Govdocs repository, a set of nearly 1 million files scraped from internet searches [27].

Table 4: Test images used during evaluation

Origin	Name	File System	Size of raw image
DigitalCorpora.org	nps-2009-domexusers	NTFS	n/a (4.1 GB E01)
Custom-made	custom-ntfs-100	NTFS	100 GB
Custom-made	custom-ext4-100	Ext4	100 GB
Custom-made	custom-ntfs-200	NTFS	200 GB
Custom-made	custom-ext4-200	Ext4	200 GB
Custom-made	custom-ntfs-250	NTFS	250 GB
Custom-made	custom-ext4-250	Ext4	250 GB

Steps taken to create custom test images:

- The custom 100 GB Ext4 image was created by downloading and unzipping files folders from the Govdocs repository. Each folder contained approximately 1000 files and the image contained approximately 190 folders.
- For the 200 GB image, the files in the 100 GB image were copied to a separate folder so that image contained duplicate files.
- The 250 GB image, was created by copying the files from the 100 GB image to the drive. Next the 100 GB file image file was compressed and stored inside of the drive. Finally, all of the zip files were included in a separate folder.
- To create the NTFS images of each size, the data from the Ext4 images was copied over using `rsync`. Each image was initially created by using `dd` from `/dev/zero` of a CentOS 7

[6] NTFS is the primary filesystem used on Windows machines, and Ext4 is the filesystem commonly found on Linux machines

VM. Next, the resulting files were attached to the VM as loop devices and `mkfs.ext4` as well as `mkfs.ntfs` were used to format the file systems.

This process resulted in each image being over 90% full. The NTFS images appeared to be less full then the Ext4 images, but it was verified that each image size had the same contents by manually inspecting the disk contents.

Table 5 gives an outline of the images that were tested on each tool. Only the custom-ntfs-100 and the nps-2009-domexusers images were tested on the AD Lab 1-DPE deployment in the interest of time. As a distributed framework, AD Lab would not be expected to perform particularly well with only one DPEGroup node; this architecture defeats the purpose of the tool and is not a fruitful exercise. Our main focus was operation with four DPEs, but we also tested some of the larger images with six and eight DPEs to explore the limits of the DPM's delegation and the resulting optimal configuration.

The custom 250 GB images were not evaluated fully on any configuration. We found during AD Lab testing that the processing often hung on custom-ntfs-250 and would not complete. This should be investigated, but as the method we used for creating the 250 GB images was not 'normal' usage, we do not extrapolate this failure to the overall ability for AD Lab to process images of this size.

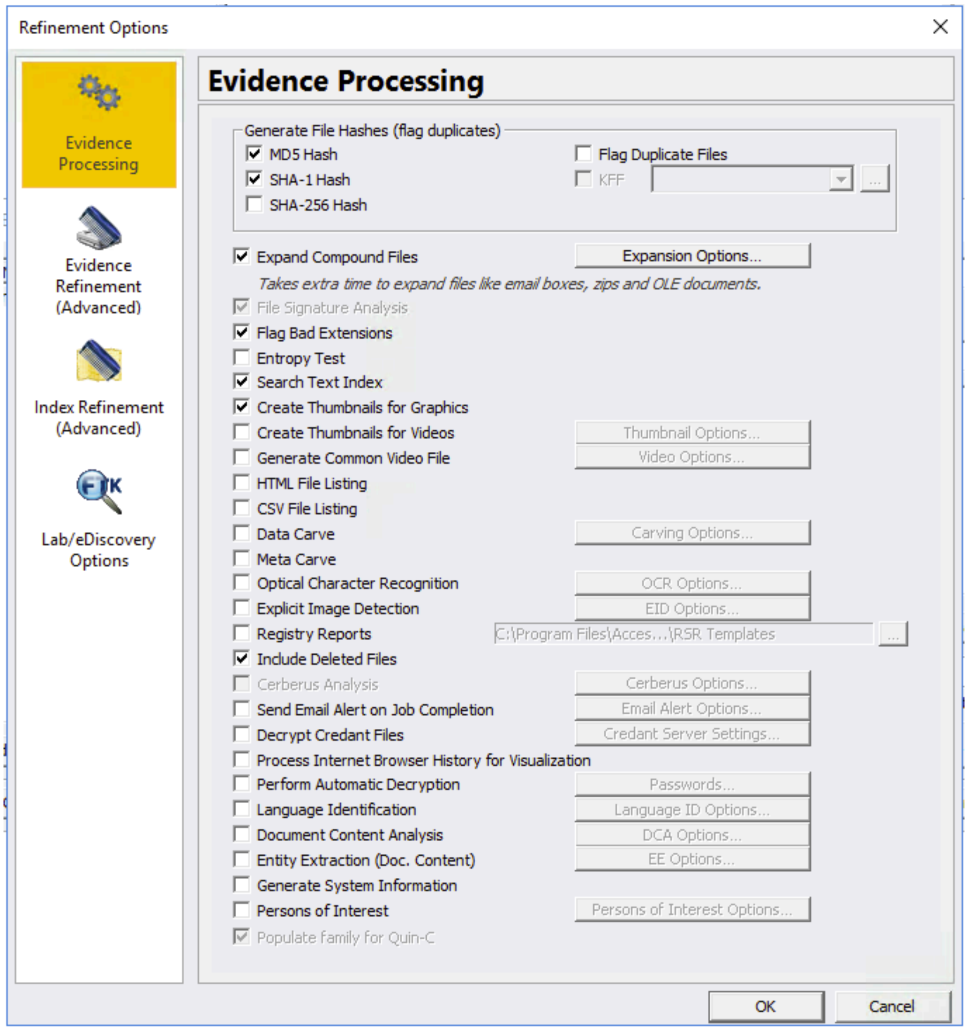
Table 5: Tests run (three full runs completed)

	AD Lab DPEs				
	1	2	4	6	8
nps-2009-domexusers	✓	✓	✓		
custom-ntfs-100	✓	✓	✓	✓	
custom-ext4-100		✓	✓		
custom-ntfs-200		✓	✓	✓	✓
custom-ext4-200		✓	✓		

5.7.3 Features Tested

In order to most accurately compare DFORC2's processing capabilities to those of AD Lab and Autopsy, our evaluation focused on a core set of features in AD Lab.

All of the tools support file hashing and indexing of strings found in files present on the hardware, and strings found in unallocated space on the device. Indexing strings during processing enables keyword searches to be performed faster in later parts of an investigation. Both of these operations' run times are directly dependent on the amount of data to be processed, and both also lend themselves to the distributed processing model; thus, they were the focus of our testing. The main configuration page of AD Lab with the options selected for this evaluation's testing is shown in Figure 1.



Refinement Options

Evidence Processing

Generate File Hashes (flag duplicates)

☑ MD5 Hash
☑ SHA-1 Hash
☐ SHA-256 Hash
☐ Flag Duplicate Files
☐ KFF

☑ Expand Compound Files
Takes extra time to expand files like email boxes, zips and OLE documents.
☑ File Signature Analysis
☑ Flag Bad Extensions
☐ Entropy Test
☑ Search Text Index
☑ Create Thumbnails for Graphics
☐ Create Thumbnails for Videos
☐ Generate Common Video File
☐ HTML File Listing
☐ CSV File Listing
☐ Data Carve
☐ Meta Carve
☐ Optical Character Recognition
☐ Explicit Image Detection
☐ Registry Reports C:\Program Files\Acces…\RSR Templates
☑ Include Deleted Files
☐ Cerberus Analysis
☐ Send Email Alert on Job Completion
☐ Decrypt Credant Files
☐ Process Internet Browser History for Visualization
☐ Perform Automatic Decryption
☐ Language Identification
☐ Document Content Analysis
☐ Entity Extraction (Doc. Content)
☐ Generate System Information
☐ Persons of Interest
☑ Populate Family for Quin-C

Figure 1: Configuration options selected during AD Lab testing for this evaluation

5.7.4 Metrics

AWS CloudWatch [28] was employed to gather metrics consistently for all AWS installations. For local installations, Linux utilities were employed to collect data. The relevant metrics identified for this evaluation are:

- Processing, Postprocessing, Indexing, and Total time
- Max CPU usage
- Max Network In/Out
- Number of findings

Note here again that the number of findings was logged for consistency of throughput comparisons only, and accuracy of findings was not evaluated. Max CPU usage and Max Network In/Out results obtained from AWS CloudWatch for AD Lab testing is the maximum reported average over a five minute period during the tests. Number of findings and processing, postprocessing, indexing, and total time metrics are those reported by AD Lab.

5.7.5 Results

In order to obtain an accurate measurement of performance, the tests shown in Table 5 were processed three times each. The results of these tests are averaged and summarized in the tables that follow. When multiple DPEs were used, the average reported is an average of all runs of all DPEs. As expected, the findings for each image tested with AD Lab were the same regardless of DPE configuration. The average total time for each test is listed in Table 6 and number of findings is summarized in Table 7.

Table 6: Results of total job time for each image in each configuration. Average of three executions (hours:minutes:seconds)

	Total Processing Time				
	AD Lab DPEs				
	1	2	4	6	8
nps-2009-domexusers	0:08:37	0:04:31	0:02:39		
custom-ntfs-100	2:43:48	1:26:33	1:05:34	1:01:02	
custom-ext4-100		1:23:00	1:01:30		
custom-ntfs-200		3:02:26	1:45:14	1:17:26	1:26:01
custom-ext4-200		2:43:45	1:42:52		

Table 7: Number of findings in each image for AD Lab

	Total Findings
	AD Lab
nps-2009-domexusers	59436
custom-ntfs-100	193982
custom-ext4-100	193018
custom-ntfs-200	387912
custom-ext4-200	386596

The maximum percent CPU utilization for each of the tests run on AD Lab shows no surprises. Larger images require a higher percentage of the CPU on both the DPM and the DPEGroup instances. Whenever more DPEGroup nodes are used, the CPU utilization on each individual node is decreased. We do notice higher CPU utilization on the DPM while processing a Ext4 image compared to the utilization demonstrated during NTFS image processing. The DPM also uses more CPU when the DPEGroup number of nodes is higher. This is also expected as the DPM must orchestrate processing across a larger number of nodes. These results are summarized in Table 8 and Table 9.

Network In/Out was the final metric collected during our testing. Table 10 shows the average maximum network in traffic in gigabytes for the DPE nodes during processing of each image, while Table 11 shows the maximum network in traffic average for the DPM. Table 12 and Table 13 show the same for network traffic out.

Table 8: Maximum Percent CPU Utilization for the DPEGroup instances as reported by AWS CloudWatch for each test configuration. Average for all DPEGroup instances over three executions. (%)

	Max Percent CPU Utilization – DPEs (%)				
	1	2	4	6	8
nps-2009-domexusers	75.0	40.9	24.6		
custom-ntfs-100	96.7	95.5	93.9	86.4	
custom-ext4-100		95.7	93.8		
custom-ntfs-200		96.4	95.1	92.2	78.5
custom-ext4-200		96.3	94.3		

Table 9: Maximum percent CPU Utilization for the DPM node as reported by AWS CloudWatch for each test configuration. Average over three executions. (%)

	Max Percent CPU Utilization – DPM (%)				
	1	2	4	6	8
nps-2009-domexusers	4.0	4.0	4.0		
custom-ntfs-100	11.0	12.3	13.4	20.3	
custom-ext4-100		24.2	22.5		
custom-ntfs-200		6.9	10.8	14.3	15.5
custom-ext4-200		18.2	23.6		

Table 10: Maximum network traffic coming in to the DPEGroup Instances as reported by AWS CloudWatch. Average over all DPEGroup instances and three executions. (GB)

	Max Network In – DPEs (GB)				
	1	2	4	6	8
nps-2009-domexusers	0.77	0.54	0.41		
custom-ntfs-100	4.50	3.52	2.76	2.38	
custom-ext4-100		3.55	2.86		
custom-ntfs-200		3.38	2.83	2.34	1.92
custom-ext4-200		3.73	2.83		

Table 11: Maximum network traffic coming in to the DPM instance as reported by AWS CloudWatch. Average over three executions. (GB)

	Max Network In – DPM (GB)				
	1	2	4	6	8
nps-2009-domexusers	0.37	0.44	0.57		
custom-ntfs-100	4.03	4.45	4.12	4.46	
custom-ext4-100		4.36	4.15		
custom-ntfs-200		3.94	4.09	5.00	4.77
custom-ext4-200		4.56	4.13		

Table 12: Maximum network traffic out of the DPEGroup instances as reported by AWS CloudWatch. Average over all DPEGroup instances and three executions. (GB)

	Max Network Out – DPEs (GB)				
	1	2	4	6	8
nps-2009-domexusers	0.37	0.21	0.21		
custom-ntfs-100	3.94	2.82	1.79	1.46	
custom-ext4-100		2.67	1.80		
custom-ntfs-200		2.69	1.81	1.65	1.33
custom-ext4-200		3.01	1.78		

Table 13: Maximum network traffic out of the DPM instance as reported by AWS CloudWatch. Average over three executions. (GB)

	Max Network Out – DPM (GB)				
	1	2	4	6	8
nps-2009-domexusers	0.84	1.13	1.53		
custom-ntfs-100	4.67	4.97	8.30	11.30	
custom-ext4-100		4.99	8.30		
custom-ntfs-200		4.53	8.61	10.60	11.97
custom-ext4-200		4.89	8.77		

5.7.6 Analysis

For our configuration of AD Lab, the increased performance attributed to using four DPEGroup nodes instead of one or two was substantial. Testing results from the 100 and 200 GB NTFS images are shown in Figure 2.

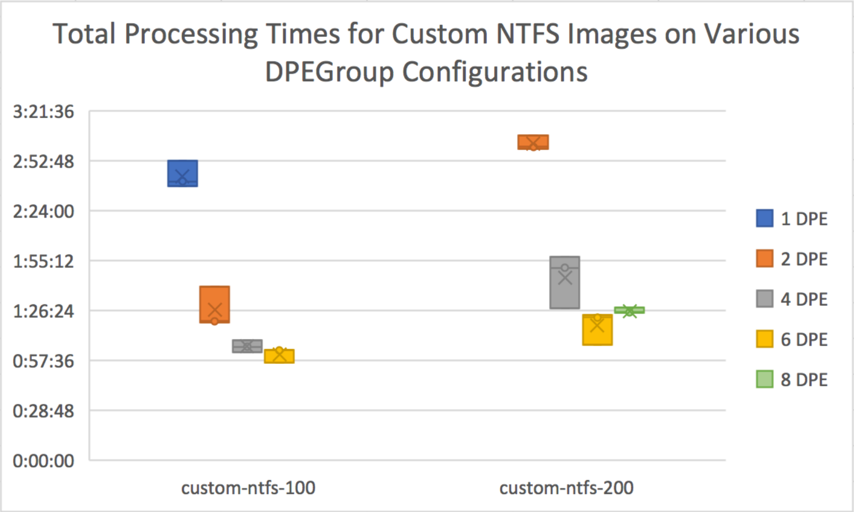


Figure 2: Comparison of AD Lab total processing time of the custom 100 and 200 GB NTFS images using 1, 2, 4, 6 and 8 DPEs. Note that the 200 GB image was not tested with 1 DPE and the 100 GB image was not tested with 8 DPEs. Notice that 6 DPEs performs better than 8 DPEs on the 200 GB image.

From one to two nodes, processing time was almost cut in half, and moving from two to four nodes saw approximately a 25% decrease in total processing time for the 100 GB images and over 35% decrease for the 200 GB images.

Six nodes performed consistently better than four nodes; however, the dramatic decrease in processing time was not as evident, only a 7% decrease on the 100 GB image and a 26% decrease on the 200 GB images tested.

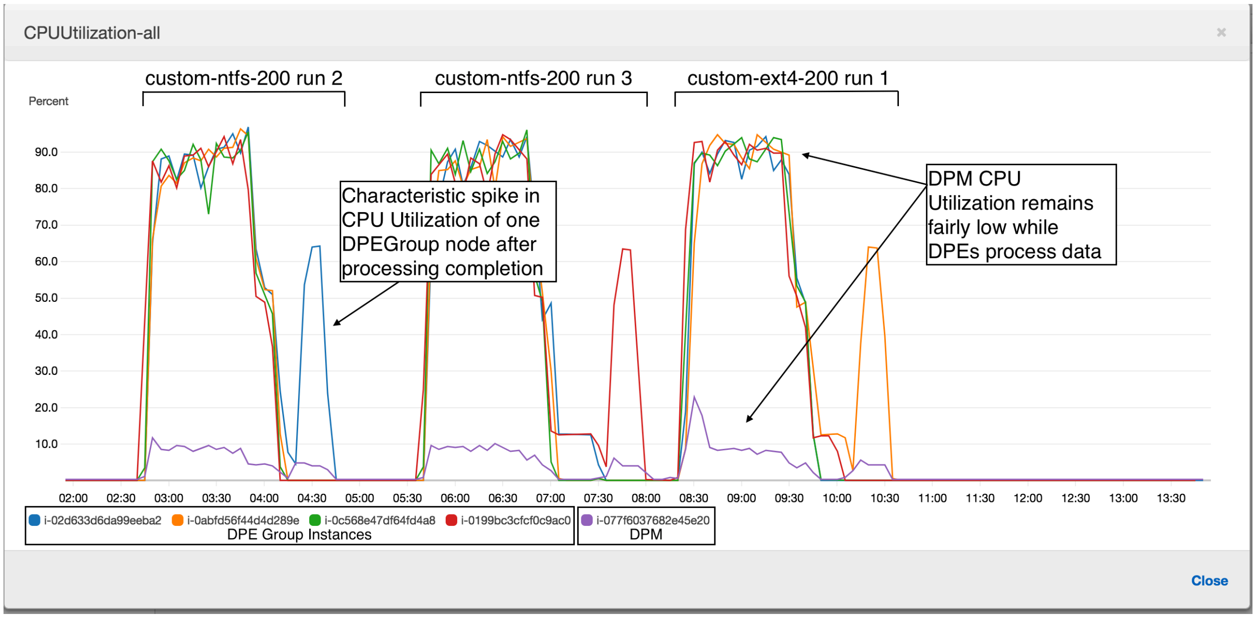
Eight nodes does not seem at all optimal; our testing witnessed a 11% increase in processing time required for the 200 GB NTFS image on eight nodes (over that of six nodes). This should be explored on more variable size images with additional testing. In addition, as noted in previous sections, with a more-capable vCPU and higher bandwidth, a more capable DPM may be better able to delegate processing to larger number of DPEs.

We did not test the 100 GB NTFS image on the eight node deployment because we witnessed the increase in time noted above, and began to see issues with DPEGroup instances being terminated and re-deployed. It is unknown if this was due to chance or the DPM orchestration of the larger number of instances. At any rate, when a node dies, due to the incomplete auto scaling capabilities in AD Lab mentioned in previous sections, the examiner must login to the DPEGroup instance manually to start services, and update the DPM to recognize the new IP address(es) of the current DPEGroup nodes. This is not ideal; processing must be stopped and the investigator must recognize that this has happened. As of this writing, no notification is sent to the investigator to trigger these actions.

In summary, with our AD Lab configurations, a six-DPE deployment seems ideal for 100- and 200-GB images with a 29% decrease in the total processing time over that of two nodes for 100 GB NTFS images, and a 58% decrease for 200 GB NTFS images. If cost is a concern, a four-DPE deployment is almost as efficient, with a 24% decrease in processing time for 100 GB NTFS images compared to that of two nodes, and a 42% decrease for 200 GB NTFS images. We did not see any noteworthy difference between processing time for NTFS images compared to Ext4 images.

In all of our tests, the DPM's CPU utilization remained below 25%. It appears to be effective at delegating processing tasks to the DPEGroup instances.

During testing, we noticed a spike in CPU utilization in one DPEGroup instance, as well as increased network traffic between that instance and the DPM in AWS after the reported completion of processing each test in AD Lab. We refrained from performing the next tests until this traffic decreased, and this data is included in the reported results for CPU utilization and network in/out in the tables provided in Section 5.7.5. It is undetermined how this traffic would affect processing of concurrent jobs. However, as results of processing were accessible during this time period and AD Lab reported completion, we did not include this additional time in the total processing time reported. Annotated examples of the CPU Utilization, Network In and Network Out, as provided by AWS CloudWatch, which demonstrate this characteristic are provided in Figure 3, Figure 4, and Figure 5, respectively.



custom-ntfs-200 run 2

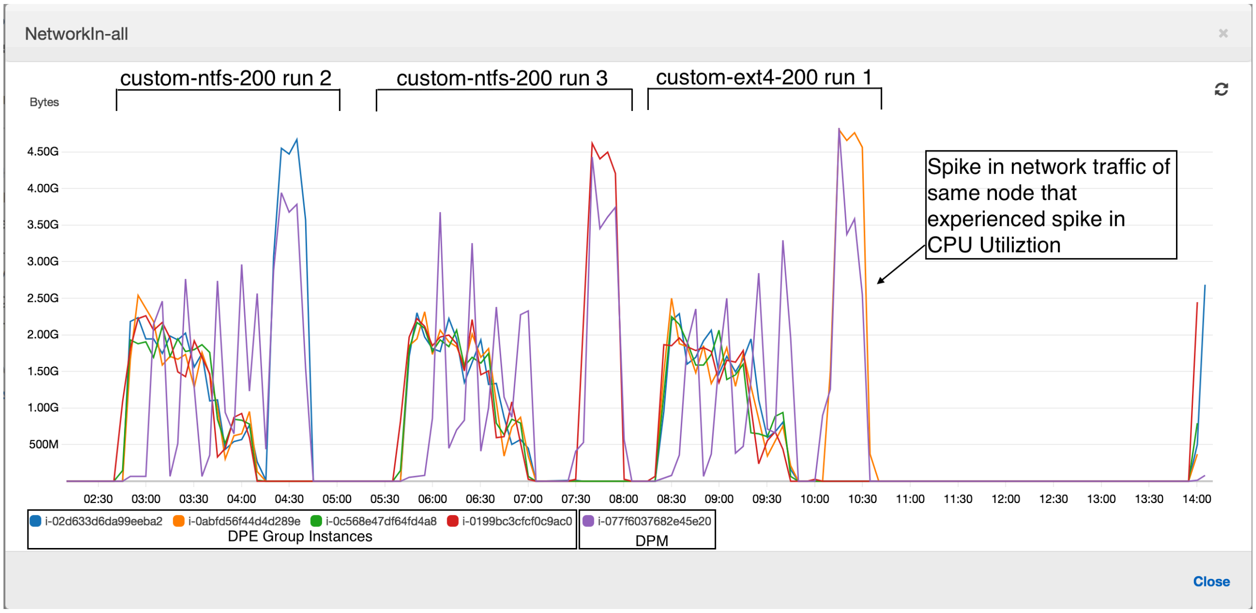
custom-ntfs-200 run 3

custom-ext4-200 run 1

Characteristic spike in CPU Utilization of one DPEGroup node after processing completion

DPM CPU Utilization remains fairly low while DPEs process data

Figure 3: Characteristic CPU Utilization for DPEGroup instances and the DPM of AD Lab. This image is an annotated screenshot of the AWS CloudWatch console showing the final two executions of the custom-ntfs-200 image with four DPEGroup instances, followed by the first execution of custom-ext4-200 on the same four DPEs.



custom-ntfs-200 run 2

custom-ntfs-200 run 3

custom-ext4-200 run 1

Spike in network traffic of same node that experienced spike in CPU Utiliztion

Figure 4: Characteristic NetworkIn metrics for DPEGroup instances and the DPM of AD Lab. This image is an annotated screenshot of the AWS CloudWatch console showing the final two executions of the custom-ntfs-200 image with four DPEGroup instances, followed by the first execution of custom-ext4-200 on the same four DPEs. On the far right, the second run of the custom-ext4-200 image is beginning.

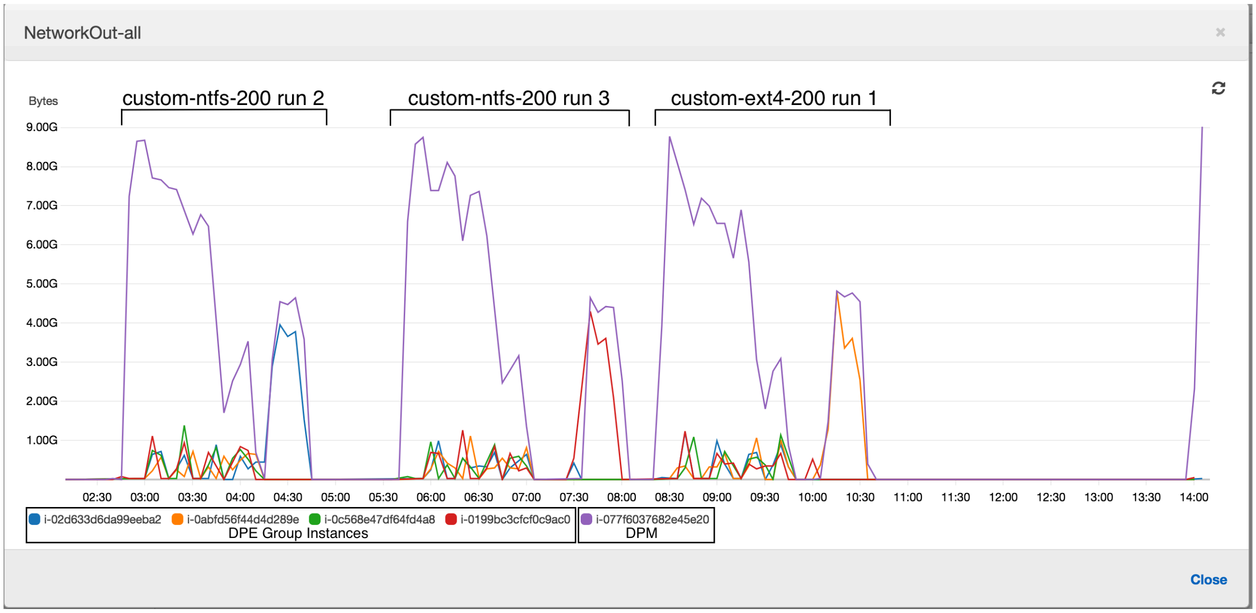


Figure 5: Characteristic NetworkOut metrics for DPEGroup instances and the DPM of AD Lab. This image is an annotated screenshot of the AWS CloudWatch console showing the final two executions of the custom-ntfs-200 image with four DPEGroup instances, followed by the first execution of custom-ext4-200 on the same four DPEs. On the far right, the second run of the custom-ext4-200 image is beginning.

During testing, we attempted to queue multiple jobs with AD Lab for processing in sequence. Although AD Lab does allow the user to queue multiple jobs for processing, it executes them in parallel. For testing purposes, this is not ideal—it eliminated the ability to gather discrete metrics for each test. As a result, we did not use this feature during testing.

The authors would like to re-iterate that the time to upload data to the AWS cloud is not included in these metrics, or the ones provided by RAND for DFORC2 [1]. The amount of time required for upload depends on various configuration options.

6. RECOMMENDATIONS

The RT&E Center's evaluation of DFORC2 over the set of requirements listed in Section 3 culminates in the following recommendations.

For investigative organizations:

- Investigative organizations or a governing body should review the guidance referenced in Section 5.1.1 to determine the correct impact level for their digital evidence data and the appropriate set of infrastructures available to their use cases.

- Organizations interested in using any of the tools evaluated herein should review the network requirements of the tool and determine if their local policy allows access to the required resources. As mentioned in Section 5.1.2, access to AWS machines from the RT&E Center's internal network required additional rules and permissions to be in place, which may be difficult to configure if the organization does not have complete administrative control over and knowledge of their network.

- Investigative organizations should consider all of the cost factors described in Section 5.4 when considering a new tool or deployment. This includes maintenance and training cost of on-premises installations, the time necessary for troubleshooting open source software, license costs, and monthly AWS fees.

Guidance for DFORC2:

- The DFORC2 codebase packaging, documentation and installation model should be improved. At the time of this writing, the tool is not in a state capable of installation without significant troubleshooting effort and advanced knowledge of the inner workings of AWS and Kubernetes. The issues enumerated in Appendix B should be addressed. DFORC2 would benefit greatly from adopting an installation model similar to that of AD Lab on the AWS Marketplace for ease of installation. Ideally, DFORC2 should also improve their AWS auto scaling capabilities to allow new nodes to deploy and join the DFORC2 installation without manual user intervention.

- DFORC2 should be modified and documentation should be provided to install DFORC2 without keeping private keys in the AWS master node of the installation.

- DFORC2 should be modified to allow AWS instances to be stopped and restarted as needed without significant hardship, in order to allow a decrease in AWS fees incurred when nodes are not in use.

- Developers of DFORC2 should continue to use static analysis tools and other software engineering practices outlined in Section 5.5.

- DFORC2 should stay up-to-date with the current versions of their dependent libraries and tools, and should identify a method to incorporate bug fixes from open source dependencies into their modified codebases.

- Although not strictly required, DFORC2 should also be further developed to be more flexible in its required infrastructure. As it stands, the tool is highly dependent on AWS and would not be portable to another cloud infrastructure without significant effort. This decreases the installation options for investigative organizations. In addition, by targeting Linux machines in code development, DFORC2 has become tied to an operating system that is not officially supported by its underlying software, Autopsy and The Sleuth Kit. Windows deployment should be considered in order to take full advantage of these tools' capabilities and future enhancements.

7. CONCLUSION

DFORC2 is a digital forensics tool prototype that builds on the open-source Autopsy software in order to enable distributed processing of digital evidence in the AWS cloud. The RT&E Center was unable to install a working version of the tool in our AWS environment without significant development effort, which was out of scope for our tasking.

Other companies and individuals are also developing products that distribute digital evidence processing in an effort to solve this same problem. Turbinia is one such tool, which takes a different approach and parallelizes the processing of multiple jobs at once. EnCase Forensic from OpenText also reports an ability to distribute processing between multiple machines, and AD Lab from AccessData takes a very similar approach to DFORC2 and distributes processing of AccessData's FTK digital forensics tool in the AWS cloud. As the most similar to DFORC2, an evaluation license for AD Lab was obtained and the tool was evaluated alongside DFORC2 during this project.

Both DFORC2 and AD Lab have the goal of engaging AWS to abstract the burden of cloud maintenance from the end user, while providing users with scalable and potentially game-changing processing capabilities. This evaluation aims to provide potential users with the knowledge required to make an informed decision on the topic of moving digital forensics processing to the cloud, and the tools available on this emerging platform.

From our interviews with practitioners, the increased processing speed reportedly acquired by these tools does address a real concern for investigators. None of the tools evaluated address the problem of uploading data to the AWS cloud. We see this step as a potential new bottleneck in future investigations as AWS digital evidence processing is adopted. Test results of AD Lab given in Section 5.7.5 as well as those reported by the developers of DFORC2 do show a significant increase in throughput for these tools.

Due to the prototype status of the DFORC2 tool and the significant development required to attain a working installation of the tool, the RT&E Center was unable to replicate the results reported by RAND for DFORC2 evidence processing. Section 6 enumerates the improvements that should be made to the DFORC2 tool in order for it to be considered generally usable among pratitioners. The architecture of the DFORC2 installation should be carefully considered in order to avoid potential security problems in the future. The tool should be updated to operate with the current versions of AWS tools and third-party software, and the installation process should be improved and documentation should be written to enable smooth installation without the need for AWS and Kubernetes expertise.

8. ABBREVIATIONS AND ACRONYMS

AD	AccessData
AD Lab	AccessData Lab
API	Application Programming Interface
AWS	Amazon Web Services
CFTT	Computer Forensics Tool Testing
CPU	Central Processing Unit
DFORC2	Digital Forensics Compute Cluster
DPE	Distributed Processing Engine
DPM	Distributed Processing Manager
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
ECU	EC2 Compute Unit
EFS	Elastic File System
Ext4	Fourth Extensible File System
FTK	Forensic Toolkit
GUI	Graphical User Interface
IP	Internet Protocol
IRB	Institutional Review Board
IT	Information Technology
ITSD	Information Technology Services Department
JHU/APL	Johns Hopkins University Applied Physics Laboratory
LEO	Law Enforcement Organization
MFA	Multifactor Authentication
NIJ	National Institute of Justice
NIST	National Institute of Standards and Technology
NSRL	National Software Reference Library
NTFS	New Technology File System
RDP	Remote Desktop Protocol
RHEL	Red Hat Enterprise Linux
RT&E Center	Research, Test, and Evaluation Center
S3	Simple Storage Service
TSK	The Sleuth Kit
vCPU	Virtual CPU
VM	Virtual Machine
VPC	Virtual Private Cloud
VPN	Virtual Private Network
YAML	YAML Ain’t Markup Language

9. BIBLIOGRAPHY

[1] D. Gonzales, Z. Winkelman, T. Tran, R. Sanchez, D. Woods and J. Hollywood, "Digital Forensics Compute Cluster: A High Speed Distributed Computing Capability for Digital Forensics," in *International Journal of Computer and Information Engineering*, Vancouver, 2017.

[2] S. L. Garfinkel, "Digital Forensics Research: the Next 10 Years," in *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, Amsterdam, 2010.

[3] S. L. Garfinkel, "Digital Media Triage with Bulk Data Analysis and Bulk Extractor," in *Computers & Security*, Oxford, 2013.

[4] "Autopsy," [Online]. Available: http://sleuthkit.org/autopsy/. [Accessed 2018].

[5] G. G. Richard III and V. Roussev, "Next-Generation Digital Forensics," in *Communications of the ACM*, New York, 2006.

[6] V. Roussev and G. G. Richard III, "Breaking the Performance Wall: The Case for Distributed Digital Forensics," in *Proceedings of the Digital Forensics Research Conference*, Baltimore, 2004.

[7] "dc3dd," SourceForge, [Online]. Available: https://sourceforge.net/projects/dc3dd/. [Accessed 2018].

[8] "Apache Kafka," The Apache Software Foundation, [Online]. Available: https://kafka.apache.org. [Accessed 2018].

[9] "Apache Spark," The Apache Software Foundation, [Online]. Available: https://spark.apache.org. [Accessed 2018].

[10] "National Software Reference Library (NSRL)," NIST, [Online]. Available: https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl. [Accessed 2018].

[11] AWS, "Previous Generation Instances," Amazon, [Online]. Available: https://aws.amazon.com/ec2/previous-generation/. [Accessed 5 June 2018].

[12] Amazon Web Services, "Amazon EC2 Pricing : On-Demand Pricing," Amazon, [Online]. Available: https://aws.amazon.com/ec2/pricing/on-demand/. [Accessed 2018].

[13] S. Mocas, "Building Theoretical Underpinnings for Digital Forensics Research," *Digital Investigation,* vol. 1, no. 1, pp. 61-68, 2004.

[14] US Department of Justice, "Searching and Seizing Computers and Obtaining Evidence in Criminal Investigations Manual," 2009.

[15] NIST, "Computer Forensics Tool Testing Program (CFTT)," NIST, [Online]. Available: https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt. [Accessed 2018].

[16] AccessData, "Forensic Toolkit (FTK)," AccessData, [Online]. Available: https://accessdata.com/products-services/forensic-toolkit-ftk. [Accessed 2018].

[17] AccessData, "AD Lab," AccessData, [Online]. Available: https://accessdata.com/products-services/ad-lab. [Accessed 2018].

[18] Google, "Turbinia," Github, [Online]. Available: https://github.com/google/turbinia. [Accessed 2018].

[19] OpenText, "EnCase Forensic," opentext, [Online]. Available: https://www.guidancesoftware.com/encase-forensic. [Accessed 2018].

[20] R. R. Varuni and R. M. Koike, "How to Use Bucket Policies and Apply Defense-in-Depth to Help Secure Your Amazon S3 Data," Amazon Web Services, 7 March 2018. [Online]. Available: https://aws.amazon.com/blogs/security/how-to-use-bucket-policies-and-apply-defense-in-depth-to-help-secure-your-amazon-s3-data/.

[21] Amazon Web Services, "AWS Snowball Pricing," Amazon, [Online]. Available: https://aws.amazon.com/snowball/pricing/. [Accessed 2018].

[22] Amazon Web Services, "Amazon S3 Pricing," Amazon, [Online]. Available: https://aws.amazon.com/s3/pricing/. [Accessed 2018].

[23] Amazon Web Services, "Amazon EBS Pricing," Amazon, [Online]. Available: https://aws.amazon.com/ebs/pricing/. [Accessed 2018].

[24] Cppcheck, "Cppcheck," SourceForge, [Online]. Available: http://cppcheck.sourceforge.net/. [Accessed 2018].

[25] SpotBugs, "SpotBugs," GitHub, [Online]. Available: https://spotbugs.github.io/.

[26] Find-Sec-Bugs, "Find Security Bugs," GitHub, [Online]. Available: https://find-sec-bugs.github.io/. [Accessed 2018].

[27] Digital Corpora, "Govdocs1," Digital Corpora, [Online]. Available: https://digitalcorpora.org/corpora/files. [Accessed 2018].

[28] Amazon Web Services, "Amazon CloudWatch," Amazon, [Online]. Available: https://aws.amazon.com/cloudwatch/. [Accessed 2018].

[29] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth Edition, USA: John Wiley & Sons, Inc., 2013.

[30] The Apache Software Foundation, "Apache ZooKeeper," The Apache Software Foundation, [Online]. Available: https://zookeeper.apache.org/. [Accessed 2018].

[31] The Apache Software Foundation, "Solr," The Apache Software Foundation, [Online]. Available: http://lucene.apache.org/solr/. [Accessed 2018].

[32] The Apache Software Foundation, "Apache ActiveMQ," The Apache Software Foundation, [Online]. Available: activemq.apache.org. [Accessed 2018].

APPENDIX A. – SECURITY GUIDANCE AND STANDARDS

A portion of our tasking focused on the cybersecurity impact of moving digital forensics processing to the cloud. This evaluation included a review of cybersecurity challenges of transitioning digital forensics operations and tools to on-premises, hybrid, or off-premises cloud infrastructures. This appendix describes the cybersecurity considerations of securing digital forensics data in a cloud environment and provides pointers to Federal, industry and vendor specific resources and recommendations to aid Federal, State and Local, Tribal and Territorial (SLTT) Law Enforcement Organizations (LEOs) in planning secure and compliant implementations of digital forensics operations in cloud based architectures with the goal for digital evidence transferred, stored or processed in a cloud environment to be accepted as admissible in court.

After outlining our scope in Section A.1, Section A.2 briefly discusses digital forensics background and key characteristics required for supporting admissibility of digital forensics data in court. Section A.3 provides an overview of select Federal guidance and programs available that focus on securing traditional and cloud infrastructures. Section A.4 describes Threat Tiers and Threat Frameworks to help determine threat actor levels, tactics and techniques used by adversaries and Section A.5 concludes the topic.

A.1 Scope and Assumptions

This cybersecurity impact evaluation of transitioning digital forensics operations to a cloud-based architecture focused on requirements, standards, resources and best practices available to LEOs to support secure processing, transit and storage of law enforcement digital forensics in a cloud environment that ultimately supports the admissibility of evidence transmitted, stored or processed in a cloud-based environment. The cloud-based environment may be a private, hybrid, community, or public cloud depending on the operating organization’s needs. References include sources for determining sensitivity of data and evaluating threats that drive protection requirements. Additional references provide focus on available security controls and benchmarks that are suitable to protect data commensurate with the sensitivity of the data and applicable threats in a private on-premises, hybrid, or off-premises cloud service. Cross-reference tables are provided to help implementers determine applicability and compliance across a range of standards.

Sources of guidance included in this document include US Federal National Institute of Standards and Technology (NIST) publications and standards, Office of Management and Budget (OMB) Federal Risk Authorization and Management Program (FedRAMP) guidance, Department of Homeland Security (DHS) Federal Critical Infrastructure (CI) protection guidance, Center for Internet Security (CIS) security benchmarks, and the Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM).

This evaluation assumes no single governing or standard body exists that establishes and maintains cybersecurity requirements and standards applicable to all LEOs. There is no one-size-fits-all approach to managing cybersecurity risk for LEOs as organizations will have unique risks and vulnerabilities, face different threats and ultimately, have different risk tolerances based on their mission, budget, sensitivity of their data and impact of compromise or loss of information technology (IT) capabilities or data. Therefore, LEOs should reference cybersecurity guidance and standards published by US Federal organizations and national and international standards bodies as well as industry to determine the best approach to protect their systems and data in both traditional and cloud infrastructures.

This section does not provide LEO requirements but offers resources to help LEOs determine appropriate security controls and solutions to protect forensics data and processing in the cloud commensurate with their unique requirements.

A.2 Background

Three cloud service delivery models are addressed in this report:

1. on-premises private cloud,
2. hybrid cloud with the infrastructure and software shared between on-premises infrastructure and off-premises cloud services, and
3. fully off-premises commercial cloud service.

In an off-premises software as a service (SaaS) model, the infrastructure is owned and managed by the commercial cloud service provider (CSP), increasing efficiency and decreasing costs as compared to traditional, on-premises data centers.

A.2.1 Digital Forensics Characteristics

Data, in any format, requires characteristics of evidence that supports its suitability or admission as fact with high confidence to persuade. In order to be useful to an investigation and prosecution, digital evidence needs to be preserved in a forensically sound manner. Acquisition of the data should be done in a manner that preserves the "complete and accurate representation of the original data while the authenticity and integrity of the data can be validated."[7]

Proper authentication demonstrates that the contents of the record remain unchanged and the information in the record originates from the source originally attributed to the data and related pertinent information such as the date and time of the record are accurate. Reliable documentation and maintenance of the chain of custody is an important aspect of supporting authenticity. A solid chain of custody provides protection against objections that the evidence was improperly handled or maintained. Additional protection is provided through applying integrity checks.

A.2.2 Admissibility Considerations for Cloud-based Digital Forensics

In the past decade, we have experienced the massive evolution of cloud computing, yet development and availability of cloud based digital forensics tools have not kept pace with availability of other cloud based applications. This has primarily been due to concerns related to

[7] E. Casey, Digital Evidence and Computer Crime, Waltham, MA: Elsevier, 2011.

security and chain of custody of forensics data in a cloud environment and the potential of a jury rejecting forensics data that had been processed or stored in the cloud[8].

Admissibility of evidence in federal court is generally governed by the Federal Rules of Evidence ("FRE"). FRE 901(b) focuses on hash values and metadata as two ways to satisfy the standard for authenticity. “The way in which software identifies, collects and stores these two areas of forensics - hash values and metadata - are most critical to determining the value of the particular software for evidence collection”[9]. FRE 902 accepts hashing as adequate integrity protection without the need for an expert witness.

It is important that forensics software: “(1) preserves the target files without alteration, (2) obtains hash values of the target files, (3) obtains copies of the target files, (4) obtains hash values for each copy, and (5) maintains the integrity of each file and hash value until admitted into evidence, including adequately documenting the chain of custody”. Evidence acquired from a live system can be vulnerable to challenge if there is a break in the chain of custody. Chain of custody of digital evidence can be supported through validation of integrity of the evidence as provided by hash value verification[8].

Integrity checks provide confidence that the evidence was not altered from collection to processing and ultimate submission as evidence in court. Integrity can be proven through the use of digital hashes and message digests. Message digest algorithms consistently produce the same result for equivalent inputs and equally important, produce a different value for different inputs. The most common algorithms used for message digests are MD5 and SHA-1[10]. Auditing controls, use of multi-factor authentication, role based access controls and encryption of data in transit and at rest are also critical factors in defending proper chain of custody. These and other critical controls are included in the NIST Special Publication 800-53 Revision 4 (NIST SP 800-53 R4), *Recommended Security Controls for Federal Information Systems*. This standard is discussed further in the following section.

A.3 Federal Guidance

Cybersecurity guidance and resources are available from a number of Federal Agencies and standards bodies including the OMB, DHS and NIST. NIST publishes the guidance for following the Federal Information Security Management Act (FISMA), while the Office of Management and Budget (OMB) governs Federal Risk and Authorization Management Program (FedRAMP), a US Federal program applicable to Federal Departments and Agencies (D/As). The program defines security requirements with tailored security controls and processes to standardize and streamline

[8] M. E. Wolf Esq., "Admissibility of Digital Evidence Derived Using WARDENT(TM)".

[9] E. Martin E. Wolf, "Admissibility of Digital Evidence Derived Using WARDEN," JHU-APL, Laurel, MD, 2018.

[10] U.S. Department of Justice Federal Bureau of Investigation Criminal Justice Information Services Division, "Security Control Mapping of CJIS Security Policy Version 5.6 Requirements to NIST Special Publication 800-53 Revision 4 06/05/2017," Federal Bureau of Investigation, 2017

the security assessment, authorization, and continuous monitoring for cloud products and services used by Federal D/As.

A.3.1 NIST Guidance

The NIST Computer Security Resource Center (CSRC) develops and publishes resources on computer, cyber, and information security and privacy. FISMA was signed into law as part of the Electronic Government Act of 2002, and defines a comprehensive framework to protect government information, operations, and assets against natural or man-made threats. NIST publishes and maintains implementation guidance for FISMA.

The Federal Information Processing Standards (FIPS) are a set of standards that describe document processing, encryption algorithms, and other information technology standards for use within government agencies and by government contractors and vendors who work with the agencies. FIPS Publication 200, *Minimum Security Requirements for Federal Information and Information Systems* is a federal standard developed by NIST to address FISMA federal mandates[11]. Federal agencies must meet the minimum security requirements through applying security controls in accordance with NIST SP 800-53 R4[12].

FIPS Publication 199, Standards for Security Categorization of Federal Information and Information Systems provides a means for organizations to categorize information and information systems as low-impact, moderate-impact, or high-impact according to their level of concern for confidentiality, integrity, and availability, and the potential impact on their assets and operations should their information and information systems be compromised through unauthorized access, use, disclosure, disruption, modification, or destruction[13]. The potential impact values assigned to the respective security objectives are the highest values among those that have been determined for each type of information resident on the information systems. If the highest value for any of the security objectives – Confidentiality, Integrity or Availability is High, then the overall impact level of the system is High. If the highest value of one is Moderate, the impact level is Moderate and if all are Low, the overall impact level of the system would be Low.

The sensitivity of digital forensics data may vary from case to case and among jurisdictions; however, if the intent is to present digital forensics data in court, the impact level for Integrity should typically be High, resulting in a High-impact level for the system. We recommend LEOs categorize their forensics information and information systems that process/store/transmit forensics data by using the FIPS 199 categorization guideline. When the need for high integrity for forensics data is required for acceptability in court, the overall impact level should be High. Section A.3.3 explains which cloud solutions meet the requirements for High impact data.

[11] NIST, "Federal Information Processing Standards Publication 200 Minimum Security Requirements for Federal Information and Information Systems," NIST, 2006.

[12] NIST, "NIST Special Publication 800-53 Revision 4 Recommended Security Controls for Federal Information Systems and Organizations," NIST, 2013.

[13] NIST, "Federal Information Processing Standard Publication 199 Standards for Security Categorization of Federal Information and Information Systems," NIST, 2004.

NIST publishes the minimum requirements for federal systems to help organizations implement FISMA. One of these key guidelines is the NIST Risk Management Framework. In April 2013, NIST published SP 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations* to aid Federal D/As in implementing FISMA to protect their IT information and systems in accordance with the impact levels derived from FIPS 199 security categorization[11,12].

NIST SP 800-53 provides a catalog of security and privacy controls to protect federal information systems and organizations. This publication also describes a process for selecting controls assets, individuals and organizational operations from a diverse set of threats. Organizations apply the appropriately-tailored set of baseline security controls to protect information and information systems commensurate with the impact level derived from FIPS 199 security categorization. Organizations have flexibility in applying the baseline security controls to tailor the relevant security control baseline to best align with their mission, business, and operational requirements[6,7].

The security and privacy controls provided are derived from Executive Orders, legislation, policies, directives, and standards, and are customizable to manage information security and privacy risk for a diverse set of organizations. These controls can then be used to implement the Risk Management Framework commensurate with the protection level required. The Framework also provides guidance on developing and applying available overlays, specialized sets of controls tailored for specific needs. For example, a Privacy Overlay is available to protect Personally Identifiable Information (PII). An organizational assessment of risk validates the initial security control selection and determines if additional controls are needed to protect organizational operations' assets or individuals. The resulting set of security controls establishes security baseline for the information system.

Identification and categorization of information systems begins with each organization conducting an organization-specific assessment of risk based on FIPS 199 and NIST SP 800-60. LEOs should consider risks to privacy, potential threats, known vulnerabilities, impact, and the overall security posture of the asset when ranking them for prioritization. The NIST SP 800 series as a whole provides a broad set of guidance on a range of additional cybersecurity topics to aid implementers in identifying and applying best security practices and standards[14].

A.3.2 DHS Guidance

DHS administers the implementation of information security policies for non-national security federal Executive Branch systems, providing policy, guidance and technical assistance. This includes developing and publishing guidance to owners and operators to protect the sixteen Critical Infrastructure (CIs) sectors. Though LEOs are not categorized as one of the sixteen CIs, LEOs may find useful guidance in DHS cybersecurity publications. For example, relevant available DHS publications include *Enabling Distributed Security in Cyberspace*, *Security Publications from US-*

[14] NIST, "Computer Security Resource Center Publication Search," NIST, [Online]. Available: https://csrc.nist.gov/publications/sp. [Accessed 2018].

CERT, *Recovering from an attack (US-CERT)*, and *Cyber Incident Reporting: A Unified Message for Reporting to the Federal Government*. These reports and more are available online[15].

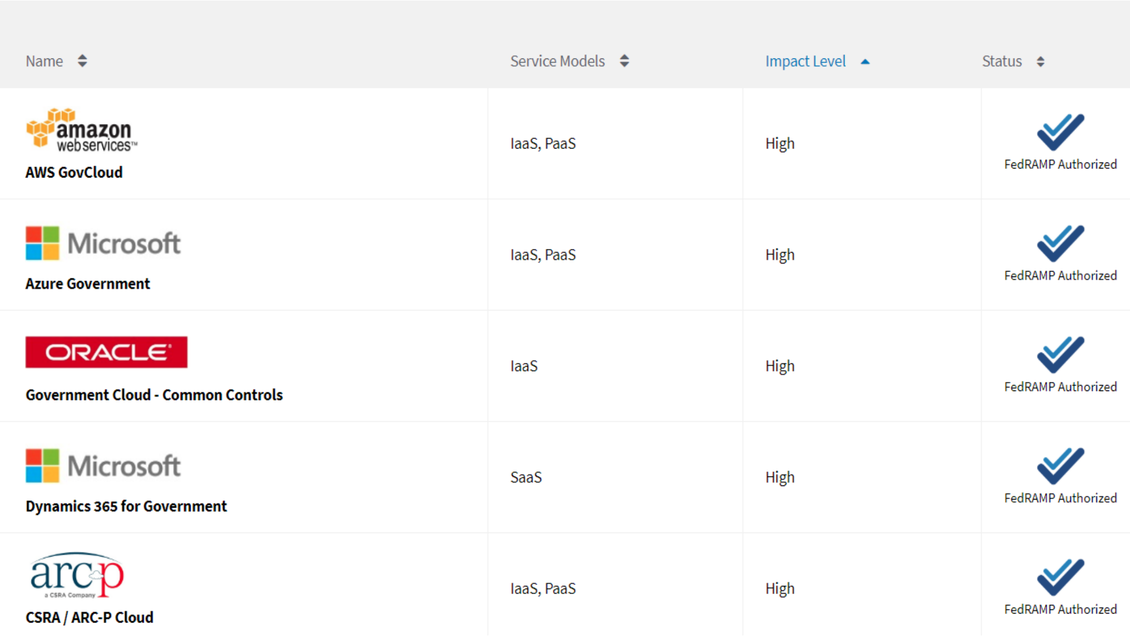
A.3.3 FedRAMP Guidance

Federal Risk and Authorization Management Program (FedRAMP) is a US Federal program applicable to Federal Departments and Agencies (D/As). The program is governed by the Office of Management and Budget (OMB) and managed by the General Services Administration (GSA). FedRAMP defines security requirements and processes to standardize and streamline the security assessment, authorization, and continuous monitoring for cloud products and services used by Federal D/As. FedRAMP assessments and authorizations are intended to be re-used throughout D/As to achieve standardization and save time, costs, and resources associated with conducting security assessment and authorizations. The FedRAMP Security Assessment Framework is based on FISMA information assurance (IA) controls implementation tailored for FedRAMP Moderate and High impact baselines. Organizations select the baseline level to protect their information and information systems in accordance with the impact level derived from their initial FIPS 199 categorization exercise.

Cloud providers must implement and have an independent third party assessor validate the NIST SP-800 controls selected for each of the baselines prior to being granted FedRAMP authorization. The FedRAMP Marketplace website lists all authorized commercial cloud solutions, their cloud service model, and impact level. The site also provides a searchable list of cloud solutions that are in the process of assessment, and those that are ready to enter the FedRAMP assessment process. The site can be sorted by assessment status (in-process, authorized), service model, deployment model, impact level, and more.

There are a limited number of CSPs that have achieved FedRAMP authorization at the High impact level. These are Amazon Web Services (AWS) GovCloud, Microsoft Azure for Government, Microsoft Dynamics 365 for Government, Oracle Common Controls for the Oracle Government Cloud - Common Controls and CSRA/ARC-P Cloud. Note that the public AWS US East/West, Microsoft Azure Commercial Cloud and Oracle Service Cloud offerings are not included in this list. These public offerings all have been granted FedRAMP Moderate impact authorizations. Figure A-1 contains an overview of authorized FedRAMP High baseline information as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS) providers.

[15] Department of Homeland Security, "DHS Cybersecurity Publications," October 2016. [Online]. Available: https://www.dhs.gov/cybersecurity-publications. [Accessed 2018].



Name	Service Models	Impact Level	Status
AWS GovCloud	IaaS, PaaS	High	✓ FedRAMP Authorized
Azure Government	IaaS, PaaS	High	✓ FedRAMP Authorized
Government Cloud - Common Controls	IaaS	High	✓ FedRAMP Authorized
Dynamics 365 for Government	SaaS	High	✓ FedRAMP Authorized
CSRA / ARC-P Cloud	IaaS, PaaS	High	✓ FedRAMP Authorized

Figure A-1: FedRAMP High Baseline Authorized Cloud Solutions (9/20/2018)[16]

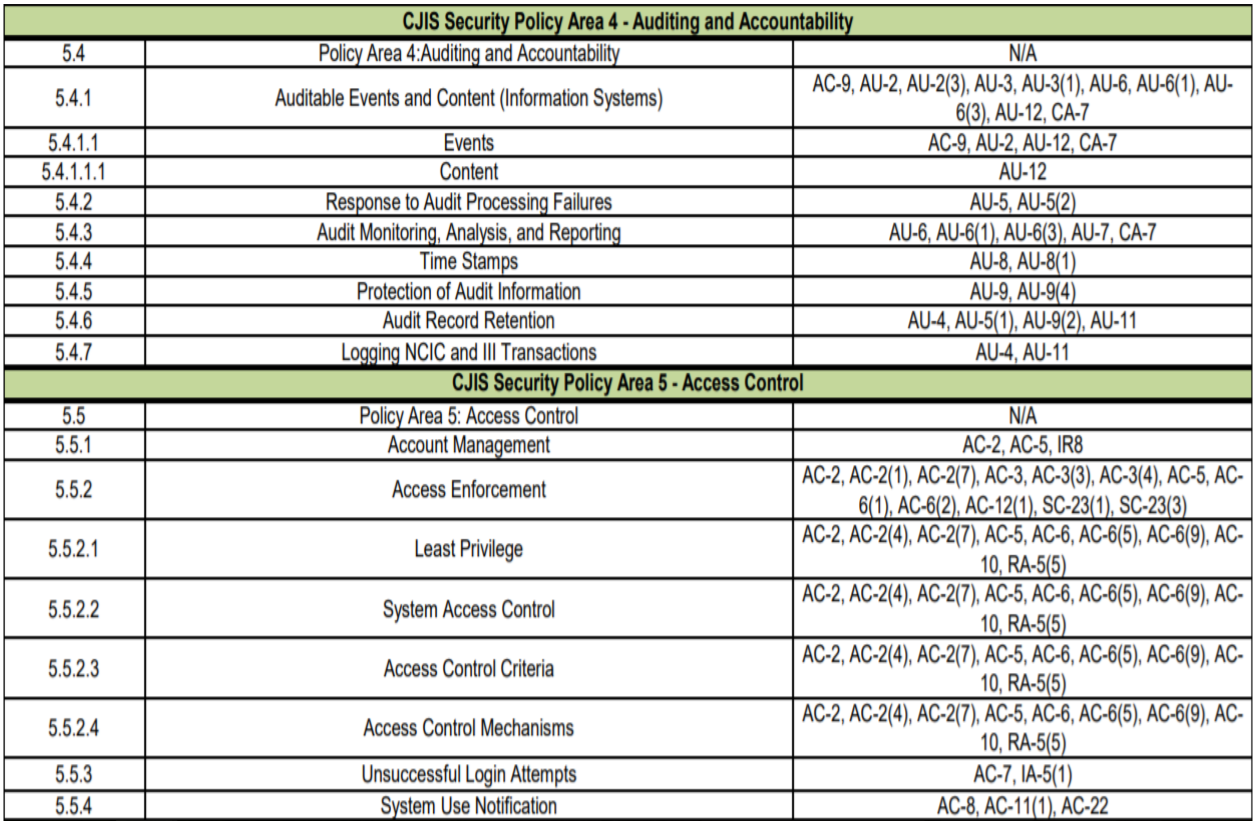
A.3.4 FBI CJIS Security Policy 2018

The Criminal Justice Information Services (CJIS) Security Policy provides security requirements, guidelines and agreements to protect sources, transmission, storage and generation of Criminal Justice Information (CJI). Since the Policy is architecture independent, an Agency can be compliant with the CJIS Security Policy when the Agency and the cloud vendor meet their respective responsibilities in the shared responsibility model to implement the controls required in the Policy.

The CJIS Security Policy provides controls appropriate to protect CJI throughout its lifecycle, including during creation, modification, while in transit, at rest and through final destruction. The policy is recommended by the criminal justice Advisory Policy Board (APB) and is based on presidential and FBI directives, federal law, and NIST guidance for FBI and CJIS Systems Agencies (CSA) implementation. The Policy provides the minimum set of security requirements and controls for access to the FBI CJIS Division systems and information and applies to all entities with access to FBI CJIS services and information. Local LEOs may develop additional policies but the CJIS Security Policy requirements set the minimum standard for protection of CJI. Section 5 of the policy provides implementation guidance for thirteen policy areas. The majority of these policy areas overlap with NIST SP 800-53 control families, including Access Control, Auditing, and Personnel Security. The Policy allows CSAs to tailor security controls commensurate with their unique risks, mission, and resources.

[16] FedRAMP PMO, "FedRAMP Products," General Services Administration, [Online]. Available: https://marketplace.fedramp.gov/#/products?status=Compliant&sort=-impactLevel. [Accessed 2018].

The FBI CJIS Information Security Officer (ISO) Program Office mapped CJIS Security Policy requirements to the security controls found in the NIST SP 800-53 R4 to provide a cross-reference between security requirements for protection of CJI and the NIST 800-53 R4 control set. Each applicable SP 800-53 R4 IA control was mapped to each CJIS policy section. Figure A-2 gives an example of the Auditing and Accountability and Access Control Security Policy Areas and their mapping to NIST SP 800-53 R4 controls. This mapping is useful for LEOs to demonstrate compliance with the CJIS Security policy through validation of proper implementation of applicable FedRAMP or FISMA NIST SP 800-53 R4 controls.



CJIS Security Policy Area 4 - Auditing and Accountability		
5.4	Policy Area 4:Auditing and Accountability	N/A
5.4.1	Auditable Events and Content (Information Systems)	AC-9, AU-2, AU-2(3), AU-3, AU-3(1), AU-6, AU-6(1), AU-6(3), AU-12, CA-7
5.4.1.1	Events	AC-9, AU-2, AU-12, CA-7
5.4.1.1.1	Content	AU-12
5.4.2	Response to Audit Processing Failures	AU-5, AU-5(2)
5.4.3	Audit Monitoring, Analysis, and Reporting	AU-6, AU-6(1), AU-6(3), AU-7, CA-7
5.4.4	Time Stamps	AU-8, AU-8(1)
5.4.5	Protection of Audit Information	AU-9, AU-9(4)
5.4.6	Audit Record Retention	AU-4, AU-5(1), AU-9(2), AU-11
5.4.7	Logging NCIC and III Transactions	AU-4, AU-11
CJIS Security Policy Area 5 - Access Control		
5.5	Policy Area 5: Access Control	N/A
5.5.1	Account Management	AC-2, AC-5, IR8
5.5.2	Access Enforcement	AC-2, AC-2(1), AC-2(7), AC-3, AC-3(3), AC-3(4), AC-5, AC-6(1), AC-6(2), AC-12(1), SC-23(1), SC-23(3)
5.5.2.1	Least Privilege	AC-2, AC-2(4), AC-2(7), AC-5, AC-6, AC-6(5), AC-6(9), AC-10, RA-5(5)
5.5.2.2	System Access Control	AC-2, AC-2(4), AC-2(7), AC-5, AC-6, AC-6(5), AC-6(9), AC-10, RA-5(5)
5.5.2.3	Access Control Criteria	AC-2, AC-2(4), AC-2(7), AC-5, AC-6, AC-6(5), AC-6(9), AC-10, RA-5(5)
5.5.2.4	Access Control Mechanisms	AC-2, AC-2(4), AC-2(7), AC-5, AC-6, AC-6(5), AC-6(9), AC-10, RA-5(5)
5.5.3	Unsuccessful Login Attempts	AC-7, IA-5(1)
5.5.4	System Use Notification	AC-8, AC-11(1), AC-22

Figure A-2: CJIS Security Policy Version 5.6 Auditing and Accountability and Access Control Security Policy Areas mapped to NIST Special Publication 800-53 R4 controls (06/05/2017)[9]

9.1.1 Center for Internet Security (CIS)

The Center for Internet Security (CIS) also publishes a list of Critical Security Controls. These are based on collective research from members. The CIS Critical Security Controls list provides a useful framework for security implementation and assessment. The controls are community developed based on real threat data across several industries and are vendor-neutral. The SANS organization publishes a mapping of CIS controls to a number of national and industry standards including NIST SP 800-53 and the Cloud Security Alliance[17].

[17] Center for Internet Security , "CIS Controls 2016 Poster," Center for Internet Security , 2016.

A.4 Threat Overview

Threats faced by LEOs will vary significantly depending on the budget, motivation, and sophistication of the tactics and techniques of threat actor groups targeting each organization. This section provides several resources for LEOs to determine their threat levels based on categorization of the threat actors targeting their organizations and tactics and techniques used by their adversaries as depicted in the threat frameworks presented in Section A.4.2. Each LEO will need to make a determination of which threat tier their most likely adversaries belong to, and plan to protect their information and information systems based on this determination.

Top Tier threats from nation state actors and well-funded highly organized criminal organizations will be highly difficult to detect and defend against. When properly implemented, NIST SP 800-53 R4 controls such as strong access controls, encryption of data-in-transit and at rest and applicable backup and recovery controls help protect data from unauthorized disclosure if compromised.

A.4.1 Defense Science Board (DSB) Threat Tiers

The Defense Science Board (DSB) categorizes threat actors into six tiers ranging from Tier I threats that rely on known exploits, to Tier VI threats originating from well trained and funded state actors groups that develop new vulnerabilities not seen before and target the supply chain of widely used commercial products. Tiers I and II are considered low tier threats, Tiers III through IV mid-tier threats and Tier V and VI are high tier threats. Tier V and VI capabilities are generally limited to a small number of countries such as the United States, China and Russia. Table A-1 provides the DSB's description of each threat tier[18].

Note that higher tier threats may use tactics and techniques available to all of the tiers below them as well. When properly implemented, defensive strategies can protect resources against Tier I and II threats; however, defending against Tier III and IV threats requires zero-day vulnerability detection tools in addition to defending against known vulnerabilities. If Tier V-VI threats are expected, deterrence needs to be part of an overall risk management strategy. Use of cloud computing can be part of this strategy as computing resources in the cloud are on-demand, providing a moving target.

[18] Department of Defense Defense Science Board, "Task Force Report: Resilient Military Systems and the Advanced Cyber Threat," Department of Defense, 2013.

Table A-1: Description of Defense Science Board Threat Tiers

Tier	Description
I	Practitioners who rely on others to develop the malicious code, delivery mechanisms, and execution strategy (use known exploits).
II	Practitioners with a greater depth of experience, with the ability to develop their own tools (from publically known vulnerabilities).
III	Practitioners who focus on the discovery and use of unknown malicious code, are adept at installing user and kernel mode root kits, frequently use data mining tools, target corporate executives and key users (government and industry) for the purpose of stealing personal and corporate data with the expressed purpose of selling the information to other criminal elements.
IV	Criminal or state actors who are organized, highly technical, proficient, well-funded professionals working in teams to discover new vulnerabilities and develop exploits.
V	State actors who create vulnerabilities through an active program to "influence" commercial products and services during design, development or manufacturing, or with the ability to impact products while in the supply chain to enable exploitation of networks and systems of interest.
VI	States with the ability to successfully execute full spectrum (cyber capabilities in combination with all of their military and intelligence capabilities) operations to achieve a specific outcome in political, military, economic, etc. domains and apply at scale.

A.4.2 Threat Frameworks

Threat frameworks and top threats published by the MITRE Corporation and the CSA can be useful to LEOs to identify threats to their organizations to help determine protection and mitigation strategies. The Adversarial Tactics Techniques and Common Knowledge (ATT&CK) framework developed by MITRE is based on eleven tactics and hundreds of techniques that attackers can leverage when attacking enterprises[19]. Every technique is based on real-world examples from malware or threat campaigns by a threat actor group and provides a reference to the published research, security blogs, or security research teams that published information on the technique. An attacker does not need to follow the tactics linearly and may use one or more tactics, or switch between them to achieve their goals. For some techniques, detailed guidance on detection or mitigation is provided that can be helpful to LEOs in vulnerability management.

A.4.3 Cloud Security Alliance (CSA) Top Threats

In February, 2016, CSA published *'The Treacherous Twelve' Cloud Computing Top Threats in 2016*, a prioritized list of cloud computing threats compiled by the CSA Top Threats working group that focused on the top twelve threats specifically related to the shared, on-demand nature of cloud computing. Architecture/design flaws related to Identity, Credential and Access Management, Insecure Application Programming Interfaces (APIs), and System and Application Vulnerabilities are included in the list of twelve below.

[19] MITRE, "MITRE ATT&CK," MITRE, [Online]. Available: https://attack.mitre.org/. [Accessed 2018].

The 'Treacherous Twelve' Cloud Computing Top Threats in 2016:

1. Data Breaches
2. Weak Identity, Credential and Access Management
3. Insecure APIs
4. System and Application Vulnerabilities
5. Account Hijacking
6. Malicious Insiders
7. Advanced Persistent Threats (APTs)
8. Data Loss
9. Insufficient Due Diligence
10. Abuse and Nefarious Use of Cloud Services
11. Denial of Service
12. Shared Technology Issues

CSA recommends using this threat report in conjunction with best practices guides, *Security Guidance for Critical Areas in Cloud Computing V.3*, *Security as a Service Implementation Guidance* and NIST Risk Management Framework guidance to manage information technology risk. More details related to these top threats are available in the full report available for download[20].

A.5 Conclusion

DFORC2 security requirements and user responsibilities vary when installed on a local single node instance on-premises, off-premises in AWS cloud or ported to a private on-premises Cloud. This section provides potential users with Federal and industry resources available to help organizations make informed decisions regarding securing digital forensics data and processes in each of the above implementation options.

A summary of high level applicable federal and industry guidance and best practices for securing data in cloud infrastructures was included in this section to aid LEOs in protecting their information and information systems commensurate with the threat to the organization, known vulnerabilities of the information systems, and potential impact of the loss of availability, confidentiality, or integrity of the data or systems. Impact to integrity of digital forensics data and tools storing, transmitting or processing forensics data should be given special consideration in determining protection controls required to support admissibility in court. Protections that support strong chain of custody proof will also be critical to successful acceptance of digital forensics data in court.

Though this document provides pointers to Federal and industry resources available for LEOs to identify and categorize their information and information systems, LEOs should continue to refine their processes and protections based on their unique lessons learned and evolution of the threat landscape applicable to their organization and operations.

[20] Top Threats Working Group, "The Treacherous 12 Cloud Computing Top Threats in 2016," Cloud Security Alliance, 2016.

A.6 Abbreviations and Acronyms

APB	Advisory Policy Board
API	Application Programming Interface
APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics Techniques and Common Knowledge (framework)
AWS	Amazon Web Services
CCM	Cloud Controls Matrix
CI	Critical Infrastructure
CIS	Center for Internet Security
CJI	Criminal Justice Information
CJIS	Criminal Justice Information System
CSA	CJIS Systems Agencies
CSA	Cloud Security Alliance
CSP	Cloud Service Provider
CSRC	Computer Security Resource Center
D/As	Departments and Agencies
DHS	Department of Homeland Security
DSB	Defense Science Board
FBI	Federal Bureau of Investigation
FedRAMP	Federal Risk Authorization Management Program
FIPS	Federal Information Processing Standard
FISMA	Federal Information Systems Management Act
FRE	Federal Rules of Evidence
GSA	General Services Administration
IA	Information assurance
IaaS	Infrastructure as a Service
ISO	Information Security Officer
IT	Information Technology
LEO	Law Enforcement Organization
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
PaaS	Platform as a Service
PII	Personally Identifiable Information
SaaS	Software as a Service
SLTT	State and Local, Tribal and Territorial
SP	Special Publications

APPENDIX B.– DFORC2 TECHNICAL OBSTACLES

This appendix enumerates the technical obstacles faced when attempting the installation of DFORC2 and recommends mitigations for several issues.

- Kubernetes Deployment
    - DFORC2 uses Kubernetes to manage containers in AWS. While RAND has made efforts to make the deployment of the Kubernetes cluster as automated as possible, issues arose that prevented it from being fully deployed. After looking through various log files on both the master and worker nodes, we determined that the `nodeup` script in the `/var/cache/kubernetes-install/` directory was failing. The cause of this is that the `docker.ce` package that is preinstalled on the selected virtual machines conflicts with the `docker-engine` package used by DFORC2. In order to stand up the complete cluster, it is necessary to purge the `docker.ce` package and install the `docker-engine` package from the `/var/cache/nodeup/packages/` directory. Once this has been accomplished, the `nodeup` script can be restarted. Only after these actions have been executed on the master and worker nodes will the cluster pass validation. From there the desh deployment script can be run.

      One of the following actions can be taken to mitigate this issue:

      1. Modify DFORC2 to use the `docker.ce` package installed on the selected images.
      2. Modify the `nodeup` script to first remove the `docker.ce` package before attempting to install the `docker-engine` package.
      3. Select an image without the `docker.ce` package installed as the base image for the master and worker nodes.

    - When deploying the containers on the Kubernetes cluster, they reside on the worker node instance groups. When the code was first received from RAND, the default number of worker nodes in the instance group was set to 3. This resulted in the repeated failure of various containers to instantiate. Through an examination of the `dforc2-startup.sh` script and various container template and YAML files we determined that several containers required more memory than was available on the node cluster. RAND was notified and the default number of nodes was increased to 10.

      We suggest the amount of memory required for the default set of containers be calculated and the default number of worker nodes be allocated automatically. Further, it would helpful to operators for DFORC2 to include example Kubernetes (`kops` and `kubectl`) commands that help determine the state of cluster nodes and container pods.

    - During evaluation, the version of `kops` used for DFORC2 was updated in the GitLab repositories. The caused role-based access control issues when attempting to deploy the cluster. This was addressed by reverting the `kops` version back to an earlier version.

- When building DFORC2 in a single machine configuration:
  - The documentation should instruct the user that `postgresql` development files are needed to satisfy `libpq-fe.h` of sleuthkit.

  - Additional packages that must also be installed include:

    - `csh` for `env_local.sh`
    - `libtool`
    - `postgresql`
    - `automake`
    - `ssh_pass`
    - `netbeans` (for building `autopsy` and the `imagetocluster` module)

  - The documentation should be updated to include:

    - `sudo yum groupinstall 'Development Tools'` for RPM-based environments
    - `sudo apt-get install build-essential` for Debian-based environment

  - When building DFORC2's modified version of Autopsy, `ant run` currently fails. This is due to the `deadlock.netbeans.org` server no longer being available. This needs to be updated to point to a different server. A bit of research revealed that the server was decommissioned toward the end of 2017. A possible work-around is to update references to this server to one of the following to receive the `task.jar` archive:

    - `http://bits.netbeans.org/8.2/fcs/uc/tasks.jar`
    - `http://bits.netbeans.org/dev/nightly/latests/uc/tasks.jar`
    - `http://updates.netbeans.org/netbeans/updates/8.2/uc/final/distribution/tasks.jar`

  - For `ant run` to complete the `nbproject/platform.properties` file in the `autopsy` folder was modified in the following manner:

    - The `bootstrap.url` was changed to one of the previously-listed `bits.netbeans.org` addresses
    - The `netbeans-patch-version` was changed to 8.0.2

  - Further when building `autopsy` and `imagetocluster` from the provided GitLab repositories, it is necessary to install `netbeans` and either update project configuration files or environmental variables to resolve error messages about the harness directory not being defined. We do not believe that this is a shortcoming of DFORC2, but a necessary step due to Autopsy being built on the netbeans platform. However, we suggest that this be stated in the documentation.

  - It should be noted that docker must be configured to use `devicemapper` storage for the Single Machine Installation to properly load the `desh-worker` image. The default option of `overlay2` will cause the image to fail when loading.

- Documentation should note that the `env_local.sh` script requires a `csh` environment. The following scripts were modified to use the IP address of virtual machines:

  - `env_local.sh`
  - `start_desh_containers.sh`
  - `start_spark_containers.sh`
  - `start_workers.sh`

  We suggest these scripts be modified to take in an environment variable to reduce the overhead and potential errors caused by repeatedly typing in the same network address.

- When running the DFORC2 tool on a single machine, it was noticed that errors regarding mismatched versions of `pg_dump` were reported and no processing occurred. We suggest the appropriate version of `pg_dump` be noted in the documentation or shipped with DFORC2 to avoid such software versioning issues.

APPENDIX C. – DFORC2 ON OPENSTACK

In this section, we provide notes and information regarding the RT&E Center's investigation of the integration of the DFORC2 software into a distributed OpenStack environment. OpenStack cloud environments are typically located on-premises and are not maintained by a third party. As such, this section begins by giving an overview of the resources necessary to deploy an OpenStack cloud. We then enumerate the additional software that must be installed on the cloud in order to support DFORC2, including some that are replacements for AWS-specific dependencies. At the beginning of this task, the RT&E Center discovered that the DFORC2 codebase is very AWS-specific, and porting the tool to OpenStack quickly became too resource-intensive to fit within the scope of this evaluation. As such, we conclude by giving an overview of the source code modifications that would have to be finished and tested for a successful deployment on OpenStack. The RT&E Center's effort on this subtask was conducted using hardware owned and maintained by JHU/APL.

For this effort, OpenStack Queens[21] was initially configured by the JHU/APL IT Services Department (ITSD) with three Dell PowerEdge servers[22] whose network was located off the JHU/APL intranet. Figure C-1 shows the hardware layout of the OpenStack servers used during the evaluation. It consisted of a controller, `a-controller`, and a storage server with three storage nodes: `a-storage1, a-storage2, and a-storage3`.

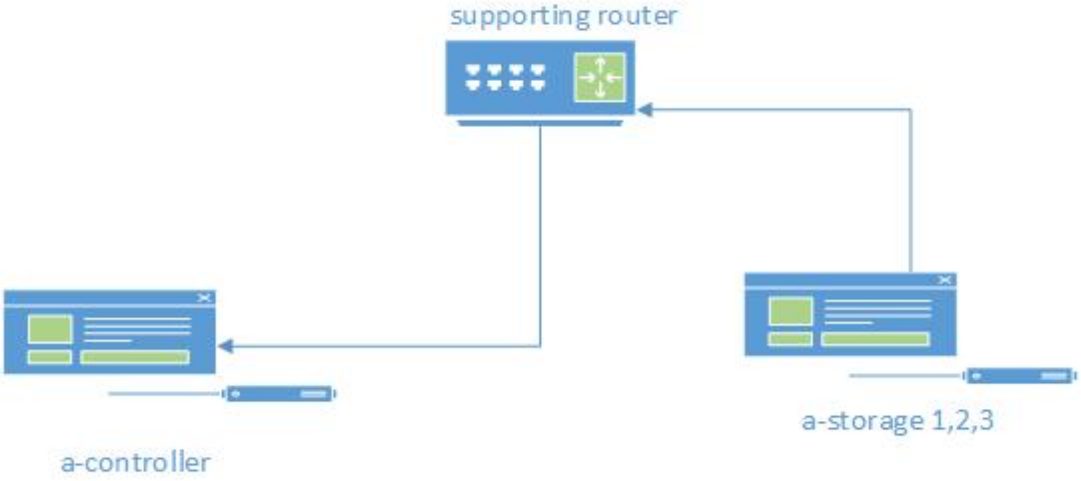


Figure C-1: OpenStack Test Lab Integration Environment

With OpenStack installed, our approach to installing the distributed DFORC2 code was to first assess the DFORC2 installation scripts and determine which OpenStack equivalent packages should be installed to meet the DFORC2 requirements. Once the initial analysis was complete, we could continue with the actual install by first installing the necessary supporting software packages. The following sections describe this process in more detail, followed by a summary and discussion of the potential of using OpenStack as an AWS equivalent in a forensics environment.

[21] OpenStack, "OpenStack Queens," OpenStack, [Online]. Available: https://www.openstack.org/software/queens/. [Accessed 2018].

[22] Dell PowerEdge 729xd with 256G RAM, 3.3T RAID 5, 1 Processor E5-2697 V3 2.6 GHz

C.1 OpenStack Equivalents of Amazon AWS Dependencies

In order to determine the packages that should be installed, we analyzed the DFORC2 scripts to determine what equivalent packages exist for the OpenStack versus an AWS environment. DFORC2 was written and designed for AWS, so some substitutions will have to be made to get it working in a OpenStack environment.

In our analysis of the DFORC2 software, we found that some of the packages installed support orchestration/direction (i.e., `default-jre`, `zookeeper`, `docker.io`, `python3-kubernetes`, `python-kafka`), while others support the storage of data. The following describes an OpenStack substitution for the AWS specified:

- `default-jre`
- ZooKeeper-related packages:
  - `zookeeperd`
  - `python-zookeeper-python-pip`
- `docker.io`
- `python3-kubernetes`
- `python-kafka`
  - Added users with `` `sudo useradd kafka -m` `` on all nodes
  - Also installed `python3-kafka` (only on `a-controller`)
- `nfs-common`
- Swift-related packages:
  - `swift`
  - `swift-proxy`
  - `memcached`
  - `python-swift-client`

On the OpenStack `a-storage` nodes, we also installed the following packages necessary to support the messaging and file system changes made with the previous set of packages:

- `xfsprogs`
- `rsync`

To set up the storage within OpenStack, mount points were configured on `/dev/sdb` and `/dev/sdc` (the `xfs` filesystem) on all storage nodes, with directories found in `/srv/node/sdb` and `/src/node/sdc`. In addition, `fstab` was updated to mount `/dev/sdb` and `/dev/sdc` at boot.

We installed OpenStack Heat to meet the requirement for a virtual private cloud (VPC)[23]. Heat has an API that can create and manage a VPC. The following Heat-related packages were installed on `a-controller`:

- `heat-common`
- `heat-api`

[23] OpenStack, "OpenStack Heat Documentation," OpenStack, [Online]. Available: https://docs.openstack.org/heat/latest/. [Accessed 2018].

- `heat-engine`
- `heat-cfntools`

When the cloud was initially installed, it included the default block storage for OpenStack. Object storage is more suitable for large files like those found in digital forensics, so the cloud was upgraded to support OpenStack's Swift object storage, which also most closely matches the S3 object storage found in AWS. Swift can be installed more easily during the initial OpenStack configuration. In future expansion of this work, this installation should be completed at this earlier phase of the process. An interface to span Amazon's S3 to the Swift object storage was also installed on all nodes where Swift was installed. This package provides an S3-like interface to the Swift object storage. DFORC2 for AWS uses S3:

- `swift-plugin-s3`

In support for the messaging requirements, we installed the library for ActiveMQ:

- `libactivemq-java`

In support of orchestration requirements, `kubernetes` and `kubectl` were installed on all nodes[24]. DFORC2 also requires the `nodeport` package. This is included with the `kubernetes` package and its related services[25].

In addition to the core packages that were installed that involve OpenStack functionality, other utilities that are needed by the DFORC2 scripts could be installed in OpenStack with the default package manager, `apt-get`:

- `curl`
- `wget`

C.2 Operating in an OpenStack Environment

Once the essential components for DFORC2 are installed in the OpenStack Linux environment, the main forensic operations of DFORC2 should be similar to those same operations in AWS. However, to get to this point involves redesign of several DFORC2 scripts. As stated previously, DFORC2 is currently very dependent on AWS utilities. One prime example of this is the installation of the object storage service Swift for OpenStack versus the S3 object storage found in AWS. Our initial testing involved setting up the DFORC2 environment, but because of its complexity, did not include fully setting up the DFORC2 forensic components in their Docker containers. The Autopsy and TSK codebases were also modified with AWS-specific calls for its

[24] The Kubernetes Authors, "Install and Set Up kubectl," The Linux Foundation, [Online]. Available: https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl. [Accessed 2018].

[25] The Kubernetes Authors, "Services," The Linux Foundation, [Online]. Available: https://kubernetes.io/docs/concepts/services-networking/service/. [Accessed 2018].

use in DFORC2. As such, additional updates may also be required in the code for the containerized objects built from these codebases.

It is possible that OpenStack can provide an equivalent distributed environment, supporting both Dockers and Kubernetes services that are essential for DFORC2. OpenStack can also support an object based storage service, Swift. Together this provides the basis for a DFORC2 install. Swift can also be configured to be accessible through an S3-equivalent system programming interface (SPI), which may ease porting of the DFORC2 tool to this infrastructure. However, to get to this point in an OpenStack installation is not a trivial task.

The OpenStack install itself is rather complex. A number of packages had to be installed to get to the base-level required in DFORC2, and modifications would have to be made to a number of DFORC2 source files. For this reason, at this point in its development, it’s not recommended for investigators attempt to deploy DFORC2 in an OpenStack environment. We recommend that only those with dedicated IT support staff well-versed in Linux and OpenStack administration attempt this installation, and only after DFORC2 formally supports OpenStack as an installation infrastructure.