2018 END-TO-END TEST DISCLOSURE AVOIDANCE SYSTEM DESIGN SPECIFICATION

Version 1.2.8

April 15, 2019

Public Release

This is version 1.2.8 (released April 2019) of the Detailed Design Specification Guidance. Printed copies of this document may not contain the most recent updates

United States™ Census Bureau

Document Revision History

Version	Publication Date	Revision Description	Section	Author(s)
1.0	10/9/2018	Initial Version	All Sections	Simson Garfinkel, Joseph Cortez
1.1	10/26/2018	Updates	All Sections	Chris Rivers
1.2	03/05/2019	Updates	All Sections	Claudia Molinar
1.2.1	03/11/2019	Updates	All Sections	Simson Garfinkel
1.2.2	03/20/2019	Updates	All Sections	Knexus Research Corporation
1.2.3	03/21/2019	Minor Edits	All Sections	Simson Garfinkel
1.2.4	03/22/2019	Addition of Appendix	All Sections	Knexus Research Corporation
1.2.4	03/25/2019	Minor Edits	All Sections	John Maron Abowd
1.2.5	03/25/2019	Updates to Appendix	All Sections	Knexus Research Corporation
1.2.6	04/01/2019	Updates to Document	All Sections	Knexus Research Corporation
1.2.7	04/05/2019	Updates to Tables	All Sections	Knexus Research Corporation
1.2.8	04/15/2019	Update to Guide	All Sections	Knexus Research Corporation

Table of Contents

List of Tables

List of Figures

1 BACKGROUND

In 2020 the United States Census Bureau will conduct the 2020 Census which aims to enumerate every person residing in the United States, covering all 50 states, the District of Columbia and Puerto Rico. All persons alive on April 1, 2020 who reside in these places, according to residency criteria finalized in 2018, must be counted, including citizens, non-citizens, people of all ages, and races and ethnic groups.

The Census Bureau must submit state population totals to the United States President by December 31, 2020. The United States Constitution mandates this decennial enumeration, used to determine each state's Congressional representation. Data from the Decennial Census is also used to aid in distributing funds from the federal to local level, and by the states for redistricting legislative districts.

As part of the Census Bureau's collection activities, the Census Bureau by statue must assure that the Decennial Census data products meet the legal requirements of Title 13, Section 9 (a)(2) of the U.S. Code, which means the published results of the census must not identify individuals, nor should individuals be reasonably inferred.

In previous Decennial Censuses a variety of techniques were used to protect the confidentiality of responses, including the use of synthetic data and household swapping.[1] For the 2020 Decennial the Census Bureau applied the latest science in developing the 2020 Disclosure Avoidance System (DAS). Following the instructions of the Data Stewardship Executive Policy Committee (DSEP), the Bureau implemented differential privacy (DP) as the primary methodology.

This public release of the 2018 End-to-End (E2E) test of the DAS source code and the accompanying release of datasets created by applying the DAS to public data from the 1940 Decennial Census provides increased transparency of the Census Bureau's effort to adopt DP at the national scale.

This document contains the following sections:

1. Background
2. General Overview
3. Disclosure Avoidance System (DAS) Operational Overview
4. DAS Infrastructure
5. DAS Design Plan
6. System Architecture
7. Cloud Based Consideration
8. Performance Metrics
9. API Interface

[1] See *Disclosure Avoidance Techniques Used for the 1970 through 2010 Decennial Censuses of Population and Housing*, Laura McKenna, October 2018, Center for Economic Studies Working Paper 18-47, US Census Bureau, available at https://www2.census.gov/ces/wp/2018/CES-WP-18-47.pdf.

2 OVERVIEW

Article 1 Section 2 of the U.S. Constitution directs the U.S. Government to conduct an "actual enumeration" of the population every ten years.

The Census Bureau will conduct the next Census of Population and Housing with reference date April 1, 2020 and produce public-use data products that conform to the requirements of Title 13 of the U.S. Code. The goal is to count everyone once, only once, and in the right place.[2] Per the code, all residents must be counted, including citizens, non-citizens, people of all ages, races and ethnic groups. After the data have been collected by the Census Bureau, but before the data are tabulated to produce data products for dissemination, the private data will undergo ***statistical disclosure limitation*** so that the impact of statistical data releases on respondent privacy can be quantified and controlled.

In the 2010 Census of Population and Housing, the trade-off between accuracy and privacy protection was viewed as a technical matter to be determined by disclosure avoidance statisticians.[3] Disclosure avoidance was performed primarily using household-level record swapping and was supported by maintaining the secrecy of key disclosure avoidance parameters.

However, there is a growing recognition in the scientific community that record-level household swapping fails to provide provable privacy guarantees. There is also growing concern that it may be possible to reconstruct a significant portion of the confidential data that underlies the census data releases using a so-called ***database reconstruction attack***, as outlined by Dinur and Nissim (2003), and that such reconstructed microdata could be used to successfully re-identify the respondents who provided a significant proportion of the underlying confidential data.

In order to fulfill its requirements to produce an accurate count and to protect personally identifiable information, the Disclosure Avoidance System for the 2020 Census will implement a new approach to disclosure avoidance that applies mathematically rigorous disclosure avoidance controls to provide the required Title 13 data protections for the released data. The Disclosure Avoidance System (DAS) will read the Census Edited File (CEF) and apply formally private algorithms to produce a Microdata Detail File (MDF). By design, the CEF will contain information that is protected by Title 13, while the MDF will not.

Thus, the DAS can be thought of as a privacy filter or barrier, that allows some aspects of data to pass while preventing leaks of Title 13 data. As an important side effect, all data that are publicly released by the Census Bureau based on the 2020 Census must go through the 2020 Disclosure Avoidance System or an alternative mathematically defensible formally private disclosure avoidance system. This includes the PL94-171 redistricting data, any summary files, quality assurance reports shared outside the Census Bureau, and other kinds of statistical summaries.

[2] https://www.census.gov/programs-surveys/decennial-census/about/why.html
[3] Note: for historical reasons, the term *disclosure avoidance* is used at the US Census Bureau to describe statistical disclosure limitation; that term will be used in the remainder of this document.

3 DAS OPERATIONAL OVERVIEW

In production, the 2020 DAS operations team launches a pre-defined Amazon Web Services (AWS) Elastic Map Reduce (EMR) cluster. This cluster includes a bootstrap script that installs pre-defined DAS code and associated software on each of the member nodes.

AWS EMR supports three kinds of nodes: a single MASTER node, CORE nodes that are used for both data storage and for computation, and TASK nodes that are only used for computation. DAS will use a single MASTER node, at least two CORE nodes, and an indeterminate number of TASK nodes.

The data transferred (file at rest) to the storage bucket attached to these nodes are encrypted using the AWS Key Management Service (KMS) utilizing 256-bit Advanced Encryption Standard (AES) encryption. Once the file is transferred to the specified S3 bucket, the master node of the cluster reads and executes the DAS code. The DAS creates an internal representation of the estimated 341 million persons and 140 million households in the U.S. in 2020. This population is arranged in a multi-dimensional national histogram (MDNH). Measurements are taken at each geographical level. Statistical noise is added to the MDNH and to the per-level measurements, privatizing the assemblage of data. Finally, the results are post-processed to create a consistent set of microdata that can be used for tabulation.

4 DAS INFRASTRUCTURE SPECIFICATION

AWS CLUSTER INSTANTIATION AND BOOTSTRAP CONFIGURATION

The DAS cluster infrastructure is a managed service provided by the Technical Integration (TI) program. The TI controls the cluster configuration which includes the AWS machine type, number of nodes for each cluster, node types for both head and core nodes, and the amount and configuration of allocated AWS Elastic Block Store (EBS) attached to each cluster node. A typical DAS baseline cluster configuration consists of the following AWS EMR cluster node specifications.

Table 1: Typical Cluster Specification - Technical Infrastructure 2020 AWS Cloud Environment

AWS Cluster Node Type	AWS EMR Cluster Role	Node Count	vCPU (Cores)	RAM (GB)	EBS Storage (GB)
MASTER	m4.16xlarge	1	6	256 GB	1000 GB
CORE	r4.16xlarge	10		488 GB	4000 GB
TASK		20			1000 GB

After the TI instantiates the cluster, a sequence of bootstrap scripts installs and configures the software necessary for DAS to execute. The first bootstrap script installs the AWS Amazon Linux (Center for Internet Security (CIS) Secure Baseline) hardened image, and installs the necessary end protection, security and monitoring tools.

After the first bootstrap script completes, a second bootstrap script installs the necessary DAS components and software tools. This bootstrap also configures the necessary cluster permissions, license configurations for the Gurobi Optimizer and other DAS-specific node configuration settings. This completes the cluster instantiation step.

The completion of the bootstrap on every node causes the DAS Step 1 script to execute on the master node which executes the DAS Core Application Framework.

ACQUIRING AND VERIFYING THE CEF

When the DAS starts the CEF has been pre-positioned in an appropriate S3 bucket at a pre-specified location. The DAS verifies that the CEF is present and properly formatted.

DAS CORE APPLICATION

DAS next executes the Top-Down DAS algorithm using the approach outlined below. This algorithm utilizes the Gurobi optimizer in parallel to generate microdata maximally consistent with a set of noisy (formally private) measurements.

First, at the national level, between 500,000 and 3,000,000 differentially private, noisy summary query measurements are taken, on which some pre-specified proportion of the global privacy budget is expended (e.g. 1/6 of total privacy budget might be spent, if the global budget is evenly split between the geographic levels).

- These summary queries are then post-processed (primarily through the solution of large-scale linear programming, quadratic programming, and mixed-integer linear/quadratic programming models constructed and solved with the Gurobi Optimizer) to generate a national-level histogram that is informationally equivalent to microdata because it is identical to the fully-saturated contingency table representation. This creates a set of non-negative, integer counts of persons and households consistent with invariants (counts to be released exactly as enumerated) agreed to by the Census Bureau's Data Stewardship Executive Policy Committee (DSEP).

- The synthetic individual and household records generated at the national level are then allocated to the 51 state-equivalent geographies using a second formally private algorithm, which again involves taking differentially private measurements (this time returning noisy counts at the state level) and generating histograms (i.e., microdata) of persons and households (but now at the state level) consistent with known invariants.[4]
- In analogous fashion, the individuals and households in each state are then allocated to counties, and then to census tracts, and then to block groups, and finally to blocks (some intermediate geographic levels may also be introduced intermediately for purely technical reasons).
- The taking of formally private measurements at each geographic level in order to make informed microdata-generation and allocation decisions consumes some portion of the privacy-loss budget.
- With each allocation, the DAS assures that several variables will be "invariant" — that is, that the tabulations of the synthetic data exactly match the tabulations of the CEF. For the 2010 Census and the 2018 End-to-End test, the invariants are listed in Table 2:
  - **C1:** Total population (invariant at the county level for the 2018 E2E)
  - **C2:** Number of housing units (invariant at the block level)
  - **C3:** Number of occupied housing units (invariant at the block level)
  - **C4:** Number of group quarters facilities by group quarters type (invariant at the block level)
- For other variables, the DAS will attempt to make the allocation of synthetic individuals match as closely as possible the actual tabulations of these variables in the CEF within the constraints allowed by the privacy-loss budgets. This goal is achieved by expending the privacy budget as wisely as is possible when taking formally private measurements at each geographic level, and then using mathematical optimization to generate microdata/allocations that closely match the noisy measurements taken.

Table 2: 2010 Census and 2018 End-to-End Test Invariants

Invariant	Definition	2010 Geographic Level	2018 End-to-End Test Geographic Level
C1	Total population	block	Providence, RI
C2	Voting-age population	block	removed
C3	Number of housing units	block	block
C4	Number of occupied housing units	block	block (note a)
C5	Number and type of group quarters	block	block (note b)
Notes: (a) DSEP recommended removing this invariant but the code base could not be adjusted in time for the test processing; (b) seven types of group quarters are invariant in the redistricting data prototype.			

END OF PROCESSING

After DAS processing, the MDF file is then written back to the specified S3 bucket, again encrypted at rest using AWS KMS. The file is then transferred to other Decennial systems for further processing.

[4] Puerto Rico is processed using the same DAS, but is given a separate privacy loss budget and is not included in totals labeled "United States."

MICRODATA FILE SPECIFICATIONS

This section of the document outlines the Decennial Census Management Division (DCMD), Center for Enterprise Dissemination - Disclosure Avoidance (CED-DA), and Decennial Information Technology Division (DITD) the specifications to create the Microdata Detail File. This specification contains the Record Layouts for the two sections of the MDF.

Table 3: Production Input

Data Title	Data File Name
Census Edited File (CEF)	CEF_COUNTS.txt CEF_PER.txt CEF_UNIT.txt
Tab GRFC	grfc_tab18_44007.txt

Table 4: Production Output

Data Title	Data File Name
Microdata Detail File (MDF)	MDF_PER.txt MDF_UNIT.txt

Table 5: Glossary and Conventions used in Record Layouts

Terminology	Definition
CENHISP	A recode of the eight edited Hispanic origin codes into 2 values representing Hispanic and not Hispanic.
CENRACE	A recode of the eight edited race codes into a single 2-digit code representing one of 63 race group categories.
CHAR(#)	A fixed-width field of # characters long. **CHAR is used for numbers if the numbers are not used for mathematical operations. CHAR is used for zero-filled numbers.**
Disclosure Avoidance (DA)	Items noted with Disclosure Avoidance (DA) have undergone disclosure avoidance in accordance with DSEP policy.
FINAL_POP	Final Population Count from the CUF – includes count imputation.
Linkage Variable	A variable that links between two tables.
INT(#)	An Integer up to # characters wide. Not zero-filled.
Not Reported	Items noted as Not Reported in the 2018 MDF represent data that might be included in the 2020 MDF but are not present in the 2018 MDF due to policy or procedural reasons. They are indicated with the notation "Not Reported"
Pipe delimited	A "pipe-delimited" file is a text file in Unicode UTF-8 encoding in which each field is separated by the Unicode Character "VERTICAL LINE" (U+007C) (e.g. "|") also known as the "pipe" character from its use in Unix pipelines.
QAGE	Edited Age as defined in the Edits and Characteristics Imputation Specification.
QRACEX	Edited Race Groups as defined in the Edits and Characteristics Imputation Specification.
QREL	Edited Relationships as defined in the Edits and Characteristics Imputation Specification.
QSEX	Edited Sex as defined in the Edits and Characteristics Imputation Specification.
Recode	A recode is a new variable that is created by combining or collapsing the value categories of an existing variable
Protected recode	A protected recode is a new variable created from **existing protected** variables.
Redundant; Remove	Items noted as Redundant exactly replicate other items already in the MDF.
RTYPE	Record Type
TEN	Edited Tenure

MICRO DATA FILE RECORD LAYOUTS

Microdata Detail File (MDF) Person Data.
Data will be pipe-delimited.
This table links to MDF.Unit using the EUID linkage variable.
This table links to GRF-C using the TABBLKST, TABBLKCOU, TABTRACTCE, TABBLKGRPCE, and TABBLK linkage variables.

Table 6: MDF.Person

#	Name	Label	Type	Values
1	**SCHEMA_TYPE_CODE**	Schema Type Code	CHAR(5)	MPD
2	**SCHEMA_BUILD_ID**	Schema Build ID	CHAR(5)	3.1.4
3	**TABBLKST**	2018 Tabulation State (FIPS)	CHAR(2)	01-02 04-06 08-13 15-42 44-51 53-56 72
4	**TABBLKCOU**	2018 Tabulation County (FIPS)	CHAR(3)	001-840 **Note:** Char(4): 0001-8400 in 2018 End-to-End
5	**TABTRACTCE**	2018 Tabulation Census Tract	CHAR(6)	000100-998999 **Note:** Not reported in 2018 End-to-End
6	**TABBLKGRPCE**	2018 Census Block Group	CHAR(1)	0-9 **Note:** Not reported in 2018 End-to-End
7	**TABBLK**	2018 Block Number	CHAR(4)	0001-9999 **Note:** Not reported in 2018 End-to-End
8	**ENUMDIST** (Note: 1940 variable only)	Enumeration District	Char(4)	0000-9999 **Note:** Only reported in 2018 End-to-End
9	**EUID**	Privacy Edited Unit ID	INT(9)	0-999999999
10	**EPNUM**	Privacy Edited Person Number	INT(5)	0-99999 **Note:** For households, EPNUM = 1 assigned to the householder (QREL = 01)
11	**RTYPE**	Record Type	CHAR(1)	3 = Person in housing unit 5 = Person in group quarters
12	**QREL**	Edited Relationship	CHAR(2)	01 = Householder 02 = Opposite-sex husband/wife/spouse 03 = Opposite-sex unmarried partner 04 = Same-sex husband/wife/spouse 05 = Same-sex unmarried partner 06 = Biological son/daughter 07 = Adopted son/daughter 08 = Stepson/stepdaughter 09 = Brother/sister 10 = Father/mother 11 = Grandchild 12 = Parent-in-law 13 = Son-in-law/daughter-in-law 14 = Other relative 15 = Housemate/roommate 16 = Foster child 17 = Other nonrelative 18 = Institutional Group Quarters Person 19 = Non-institutional Group Quarters Person 99 = Not reported in 2018 End-to-End
13	**QSEX**	Edited Sex	CHAR(1)	1 = Male 2 = Female 9 = Not reported in 2018 End-to-End
14	**QAGE**	Edited Age	INT(3)	0-115 **Note:** For 2018 End-to-End: QAGE = 17 assigned to minors, and QAGE = 18 assigned to voting age persons
15	**CENHISP**	Hispanic Origin	CHAR(1)	1 = Not Hispanic 2 = Hispanic

16	**CENRACE**	Census Race	CHAR(2)	01 = White alone 02 = Black alone 03 = AIAN alone 04 = Asian alone 05 = NHPI alone 06 = SOR alone 07 = White; Black 078 = White; AIAN 09 = White; Asian 10 = White; NHPI 11 = White; SOR 12 = Black; AIAN 13 = Black; Asian 14 = Black; NHPI 15 = Black; SOR 16 = AIAN; Asian 17 = AIAN; NHPI 18 = AIAN; SOR 19 = Asian; NHPI 20 = Asian; SOR 21 = NHPI; SOR 22 = White; Black; AIAN 23 = White; Black; Asian 24 = White; Black; NHPI 25 = White; Black; SOR 26 = White; AIAN; Asian 27 = White; AIAN; NHPI 28 = White; AIAN; SOR 29 = White; Asian; NHPI 30 = White; Asian; SOR 31 = White; NHPI; SOR 32 = Black; AIAN; Asian 33 = Black; AIAN; NHPI 34 = Black; AIAN; SOR 35 = Black; Asian; NHPI 36 = Black; Asian; SOR 37 = Black; NHPI; SOR 38 = AIAN; Asian; NHPI 39 = AIAN; Asian; SOR 40 = AIAN; NHPI; SOR 41 = Asian; NHPI; SOR 42 = White; Black; AIAN; Asian 43 = White; Black; AIAN; NHPI 44 = White; Black; AIAN; SOR 45 = White; Black; Asian; NHPI 46 = White; Black; Asian; SOR 47 = White; Black; NHPI; SOR 48 = White; AIAN; Asian; NHPI 49 = White; AIAN; Asian; SOR 50 = White; AIAN; NHPI; SOR 51 = White; Asian; NHPI; SOR 52 = Black; AIAN; Asian; NHPI 53 = Black; AIAN; Asian; SOR 54 = Black; AIAN; NHPI; SOR 55 = Black; Asian; NHPI; SOR 56 = AIAN; Asian; NHPI; SOR 57 = White; Black; AIAN; Asian; NHPI 58 = White; Black; AIAN; Asian; SOR 59 = White; Black; AIAN; NHPI; SOR 60 = White; Black; Asian; NHPI; SOR 61 = White; AIAN; Asian; NHPI; SOR 62 = Black; AIAN; Asian; NHPI; SOR 63 = White; Black; AIAN; Asian; NHPI; SOR **Note:** Only values 0 through 6 reported in 2018 End-to-End
17	**QSPANX**	Edited Hispanic Origin Group	CHAR(4)	9999 = Not reported in 2018 End-to-End
18	**QRACE1**	Edited Detailed Race 1	CHAR(4)	9999 = Not reported in 2018 End-to-End
19	**QRACE2**	Edited Detailed Race 2	CHAR(4)	9999 = Not reported in 2018 End-to-End
20	**QRACE3**	Edited Detailed Race 3	CHAR(4)	9999 = Not reported in 2018 End-to-End
21	**QRACE4**	Edited Detailed Race 4	CHAR(4)	9999 = Not reported in 2018 End-to-End

22	**QRACE5**	Edited Detailed Race 5	CHAR(4)	9999 = Not reported in 2018 End-to-End
23	**QRACE6**	Edited Detailed Race 6	CHAR(4)	9999 = Not reported in 2018 End-to-End
24	**QRACE7**	Edited Detailed Race 7	CHAR(4)	9999 = Not reported in 2018 End-to-End
25	**QRACE8**	Edited Detailed Race 8	CHAR(4)	9999 = Not reported in 2018 End-to-End
26	**CIT**	Citizenship	CHAR(1)	9 = Not reported in 2018 End-to-End

Microdata Detail File (MDF) Unit Data.
Data will be pipe-delimited.
This table links to MDF.Person using the EUID linkage variable.
This table links to GRF-C using the TABBLKST, TABBLKCOU, TABTRACTCE, TABBLKGRPCE, and TABBLK linkage variables.

Table 7: MDF.Unit

#	Name	Label	Type	Values
1	**SCHEMA_TYPE_CODE**	Schema Type Code	CHAR(5)	MUD
2	**SCHEMA_BUILD_ID**	Schema Build ID	CHAR(5)	3.1.4
3	**TABBLKST**	2018 Tabulation State (FIPS)	CHAR(2)	01-02 04-06 08-13 15-42 44-51 53-56 72
4	**TABBLKCOU**	2018 Tabulation County (FIPS)	CHAR(3)	001-840 **Note:** Char(4): 0001-8400 in 2018 End-to-End
5	**TABTRACTCE**	2018 Tabulation Census Tract	CHAR(6)	000100-998999 **Note:** Not reported in 2018 End-to-End
6	**TABBLKGRPCE**	2018 Census Block Group	CHAR(1)	0-9 **Note:** Not reported in 2018 End-to-End
7	**TABBLK**	2018 Block Number	CHAR(4)	0001-9999 **Note:** Not reported in 2018 End-to-End
8	**ENUMDIST** (Note: 1940 variable only)	Enumeration District	Char(4)	0000-9999 **Note:** Only reported in 2018 End-to-End
9	**EUID**	Privacy Edited Unit ID	INT(9)	0-999999999
10	**RTYPE**	Record Type	CHAR(1)	2 = Housing unit 4 = Group quarters
11	**GQTYPE**	Group Quarters Type	CHAR(3)	000 = NIU 101 = Federal detention centers 102 = Federal prisons 103 = State prisons 104 = Local jails and other municipal confinement facilities 105 = Correctional residential facilities 106 = Military disciplinary barracks and jails 201 = Group homes for juveniles (non-correctional) 202 = Residential treatment centers for juveniles (non-correctional) 203 = Correctional facilities intended for juveniles 301 = Nursing facilities/skilled nursing facilities 401 = Mental (psychiatric) hospitals and psychiatric units in other hospitals 402 = Hospitals with patients who have no usual home elsewhere 403 = In-patient hospice facilities 404 = Military treatment facilities with assigned patients 405 = Residential schools for people with disabilities 501 = College/university student housing (college/university owned/leased/managed) 502 = College/university housing (privately owned/leased/managed) 601 = Military quarters 602 = Military ships

				701 = Emergency and transitional shelters (with sleeping facilities) for people experiencing homelessness 702 = Soup kitchens 704 = Regularly scheduled mobile food vans 706 = Targeted non-sheltered outdoor locations 801 = Group homes intended for adults 802 = Residential treatment centers for adults 900 = Maritime/merchant vessels 901 = Workers' group living quarters and job corps centers 903 = Living quarters for victims of natural disasters 904 = Religious group quarters and domestic violence shelters **Note:** For 2018 End-to-End, only the first digit is reported.
12	**TEN**	Tenure	CHAR(1)	0 = NIU 1 = Owned with a mortgage 2 = Owned free and clear 3 = Rented 4 = Occupied without payment of rent 9 = Occupied unit in 2018 End-to-End
13	**VACS**	Vacancy Status	CHAR(1)	0 = NIU 1 = Vacant, for rent 2 = Vacant, rented, not occupied 3 = Vacant, for sale only 4 = Vacant, sold, not occupied 5 = Vacant, for seasonal, recreational, or occasional use 6 = Vacant, for migrant workers 7 = Vacant, other 9 = Vacant unit in 2018 End-to-End
14	**FINAL_POP**	Population Count	INT(5)	0-99999 **Note:** Int(2): 0-99 in 1940 data
15	**HHT**	Household/Family Type	CHAR(1)	0 = NIU 1 = Married couple household 2 = Other family household: Male householder 3 = Other family household: Female householder 4 = Nonfamily household: Male householder, living alone 5 = Nonfamily household: Male householder, not living alone 6 = Nonfamily household: Female household, living alone 7 = Nonfamily household: Female household, not living alone 9 = Not reported in 2018 End-to-End

16	**HHT2**	Household/Family Type (NEW)	CHAR(2)	00 = NIU 01 = Married couple household: With own children < 18 02 = Married couple household: No own children < 18 03 = Cohabiting couple household: With own children < 18 04 = Cohabiting couple household: No own children < 18 05 = Female householder, no spouse/partner present: Living alone 06 = Female householder, no spouse/partner present: With own children < 18 07 = Female householder, no spouse/partner present: With relatives, no own children < 18 08 = Female householder, no spouse/partner present: Only nonrelatives present 09 = Male householder, no spouse/partner present: Living alone 10 = Male householder, no spouse/partner present: With own children < 18 11 = Male householder, no spouse/partner present: With relatives, no own children < 18 12 = Male householder, no spouse/partner present: Only nonrelatives present 99 = Not reported in 2018 End-to-End
17	**NPF**	Number of People in Family	INT(2)	0 = NIU 2-97 99 = Not reported in 2018 End-to-End
18	**CPLT**	Couple Type	CHAR(1)	0 = NIU 1 = Opposite-sex husband/wife/spouse household 2 = Same-sex husband/wife/spouse household 3 = Opposite-sex unmarried partner household 4 = Same-sex unmarried partner household 9 = Not reported in 2018 End-to-End
19	**UPART**	Presence and Type of Unmarried Partner Household	CHAR(1)	0 = NIU 1 = Male householder and male partner 2 = Male householder and female partner 3 = Female householder and female partner 4 = Female householder and male partner 5 = All other households 9 = Not reported in 2018 End-to-End
20	**MULTG**	Multigenerational Household	CHAR(1)	0 = NIU 1 = Not a multigenerational household 2 = Yes, a multigenerational household 9 = Not reported in 2018 End-to-End
21	**HHLDRAGE**	Age of Householder	INT(3)	0 = NIU 15-115 999 = Not reported in 2018 End-to-End
22	**HHSPAN**	Hispanic Householder	CHAR(1)	0 = NIU 1 = Not Hispanic 2 = Hispanic 9 = Not reported in 2018 End-to-End

23	**HHRACE**	Race of Householder	CHAR(2)	00 = NIU 01 = White alone 02 = Black alone 03 = AIAN alone 04 = Asian alone 05 = NHPI alone 06 = SOR alone 07 = White; Black 08 = White; AIAN 09 = White; Asian 10 = White; NHPI 11 = White; SOR 12 = Black; AIAN 13 = Black; Asian 14 = Black; NHPI 15 = Black; SOR 16 = AIAN; Asian 17 = AIAN; NHPI 18 = AIAN; SOR 19 = Asian; NHPI 20 = Asian; SOR 21 = NHPI; SOR 22 = White; Black; AIAN 23 = White; Black; Asian 24 = White; Black; NHPI 25 = White; Black; SOR 26 = White; AIAN; Asian 27 = White; AIAN; NHPI 28 = White; AIAN; SOR 29 = White; Asian; NHPI 30 = White; Asian; SOR 31 = White; NHPI; SOR 32 = Black; AIAN; Asian 33 = Black; AIAN; NHPI 34 = Black; AIAN; SOR 35 = Black; Asian; NHPI 36 = Black; Asian; SOR 37 = Black; NHPI; SOR 38 = AIAN; Asian; NHPI 39 = AIAN; Asian; SOR 40 = AIAN; NHPI; SOR 41 = Asian; NHPI; SOR 42 = White; Black; AIAN; Asian 43 = White; Black; AIAN; NHPI 44 = White; Black; AIAN; SOR 45 = White; Black; Asian; NHPI 46 = White; Black; Asian; SOR 47 = White; Black; NHPI; SOR 48 = White; AIAN; Asian; NHPI 49 = White; AIAN; Asian; SOR 50 = White; AIAN; NHPI; SOR 51 = White; Asian; NHPI; SOR 52 = Black; AIAN; Asian; NHPI 53 = Black; AIAN; Asian; SOR 54 = Black; AIAN; NHPI; SOR 55 = Black; Asian; NHPI; SOR 56 = AIAN; Asian; NHPI; SOR 57 = White; Black; AIAN; Asian; NHPI 58 = White; Black; AIAN; Asian; SOR 59 = White; Black; AIAN; NHPI; SOR 60 = White; Black; Asian; NHPI; SOR 61 = White; AIAN; Asian; NHPI; SOR 62 = Black; AIAN; Asian; NHPI; SOR 63 = White; Black; AIAN; Asian; NHPI; SOR 99 = Not reported in 2018 End-to-End
24	**PAOC**	Presence and Age of Own Children Under 18	CHAR(1)	0 = NIU 1 = With own children under 6 year only 2 = With own children 6-17 years only 3 = With own children under 6 years and 6-17 years 4 = No own children

				9 = Not reported in 2018 End-to-End
25	**P18**	Number of People Under 18 Years in Household	INT(2)	0 = NIU 1-97 99 = Not reported in 2018 End-to-End
26	**P60**	Number of People 60 Years and Over in Household	INT(2)	0 = NIU 1-97 99 = Not reported in 2018 End-to-End
27	**P65**	Number of People 65 Years and Over in Household	INT(2)	0 = NIU 1-97 99 = Not reported in 2018 End-to-End
28	**P75**	Number of People 75 Years and Over in Household	INT(2)	0 = NIU 1-97 99 = Not reported in 2018 End-to-End

5 DESIGN PLAN SPECIFICATIONS

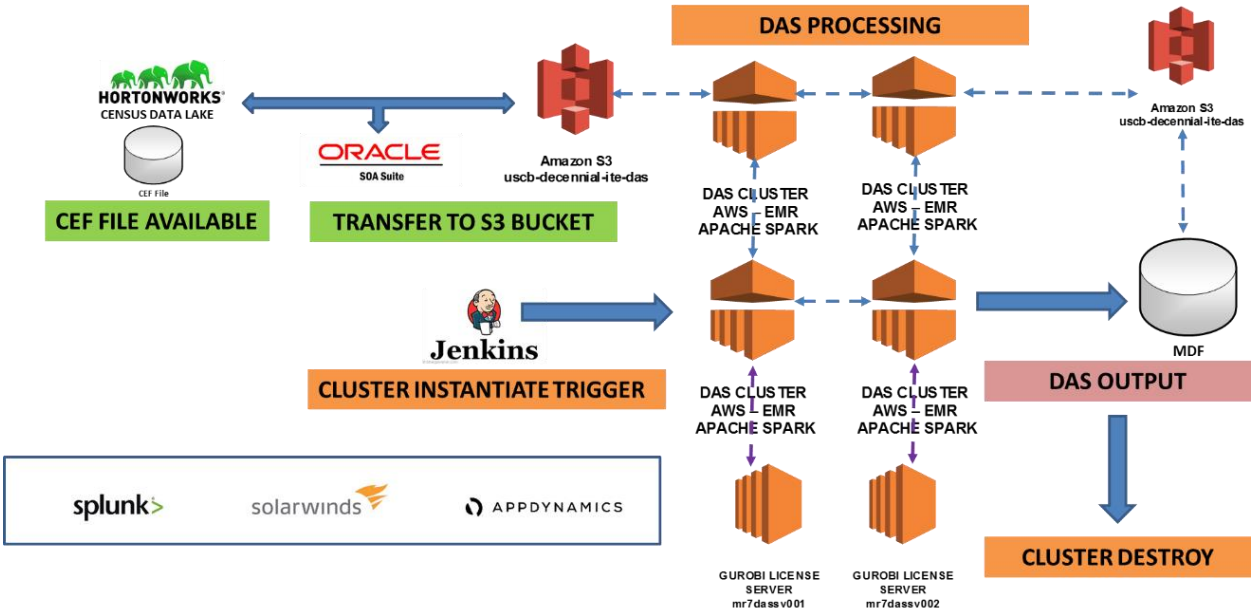
The DAS design plan can be described by the following components:

DAS CLUSTER INFRASTRUCTURE

DAS utilizes AWS GovCloud and the following AWS components:

- AWS Elastic Map Reduce (EMR) Cluster installed with Apache Spark
- AWS Simple Storage Service (S3)
- AWS Simple Notification Service (SNS)
- AWS Elastic Block Store (EBS)
- AWS Elastic Compute Cloud (EC2)

Figure 1: DAS Cluster Infrastructure



DAS ⇔ S3 INTERFACE FRAMEWORK

DAS receives the Census Edited File (CEF) from S3 writes back the results as a Micro Data File (MDF).

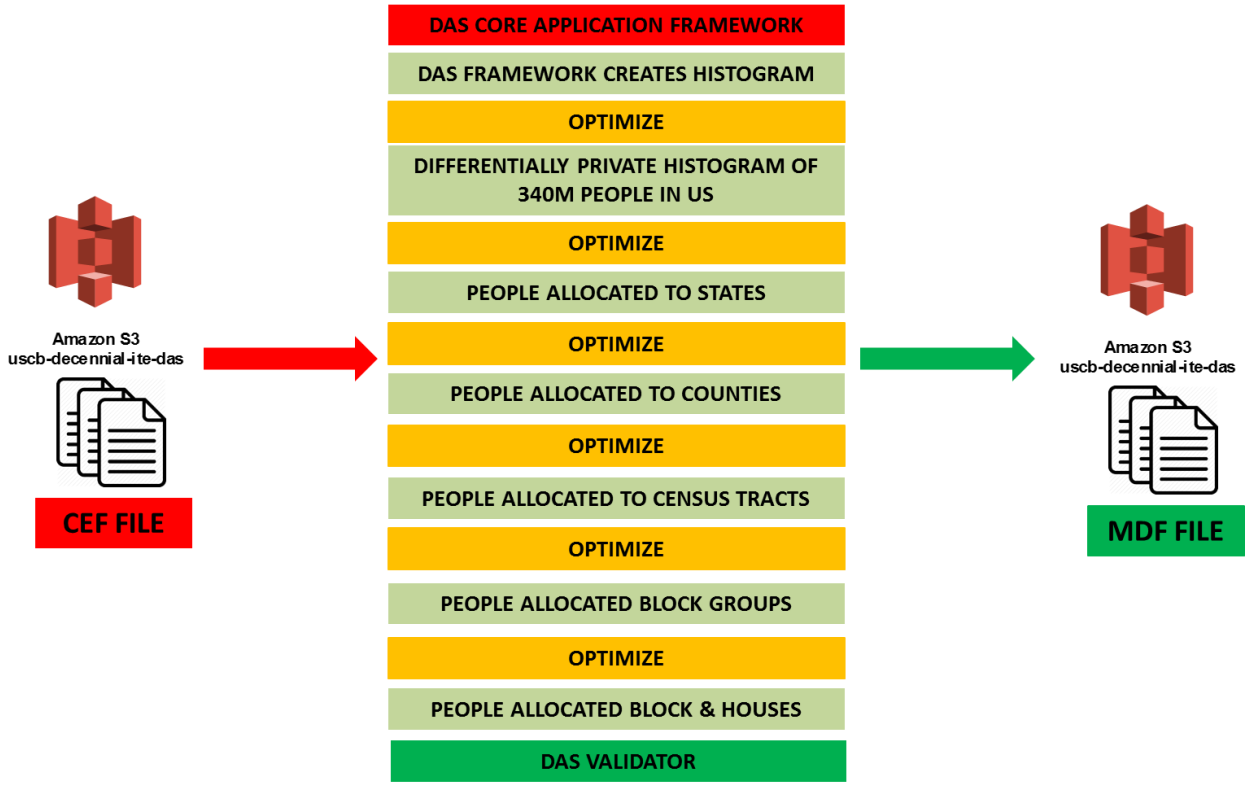
DAS CORE APPLICATION FRAMEWORK

The Disclosure Avoidance System (DAS) applies privacy controls to microdata in the data flow from the Census Edited File (CEF) to the Microdata Detail File (MDF). The privacy controls assure that there is no direct mapping between individual records in the CEF to individual records in the MDF and regulate the privacy loss implied by the production of the MDF.

Following the application of the privacy controls, the microdata in the MDF is ready for tabulation. By design, the MDF does not need to be protected under Title 13, and could, for example, be publicly released.

The CEF contains other private information that is not destined for the MDF to support other Census business processes. Gurobi optimization software is used by DAS for mathematical optimization (typically to generate or allocate microdata from noisy measurements).

Figure 2: DAS Core Application Framework



6 SYSTEM ARCHITECTURE

The section below provides an overview of the DAS System Architecture.

BUSINESS ARCHITECTURE

DAS is solving a particular data-processing problem for the Census Bureau, that is to apply disclosure avoidance and privacy controls to uphold Title 13 mandated confidentiality protections to the data published from the 2020 Census. DAS is not integrated into, nor solving any Census Bureau operational, business, financial, or transactional functions.

APPLICATION ARCHITECTURE (FRONT END)

DAS is designed and built without user access. DAS employs a single DAS operations account to run DAS from start to completion. No front-end user interface is built into DAS.

INFORMATION ARCHITECTURE (DATA)

DAS is designed and architected without any user data input, data transactions or data storage requirements. DAS will not employ a database engine in order to complete the system's intended use case.

DATA-AT-REST SECURITY CONTROLS:

CLUSTER NODE(S) ATTACHED ELASTIC BLOCK STORAGE (EBS)

When an ***encrypted*** Amazon EBS volume is attached to a supported Amazon Elastic Map Reduce (EMR) instance, data stored at rest on the volume, disk I/O, and snapshots created from the volume are all encrypted. Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys (CMKs) when creating encrypted volumes and any snapshots created from them. The encryption occurs on the servers that host Amazon Elastic Map Reduce (EMR) cluster node members. When the DAS EMR cluster is instantiated an ***encrypted EBS volume*** is automatically attached to all the node members, as a result the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

S3 BUCKET DATA ENCRYPTION

The TI 2020 Cloud supports AWS S3 bucket integration for AWS EMR. Every cluster that requires S3 storage will be assigned a specific S3 storage bucket, restricted to each project cluster node members and configured to encrypt any file stored to the assigned bucket. Amazon S3 encryption provides a way to set the encryption behavior for an S3 bucket. DAS AWS EMR sets encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption AWS KMS-managed keys (SSE-KMS).

DATA-IN-FLIGHT SECURITY CONTROLS:

Clusters instantiated within the TI 2020 Cloud, will be installed with TLS certificates for node-to-node communications for EMR-specific task execution.

BASELINE MANAGEMENT

Architecture, infrastructure and code baseline management will adhere to the TI-2020 Program Change Management Plan.

7 CLOUD BASED SECURITY CONSIDERATIONS

Table 8: Cloud Based Security Considerations

Consideration	Response
Uptime expectation from the Business Owner of the System in the Cloud:	100%
Cloud Based SLA for security monitoring:	100%
Cloud Based System Level High Availability:	YES
Cloud Based Site Level Disaster Recovery for the System:	Provided TI 2020 Cloud Capability
State whether the Cloud Vendor is FEDRAMP Certified:	YES
Data Retention Requirements:	YES

8 RELIABILITY, MAINTAINABILITY AND AVAILABILITY CONSIDERATIONS

The design of the system is such that there is a single MASTER node and multiple WORKER and/or CORE nodes. If a WORKER/CORE node fails, the EMR system will restart that load and schedule work one the failed node to be re-computed on a new node. However, there are no provisions in EMR for a failed MASTER node. If the MASTER node fails, the system will need to be manually restarted.

The current design has minimal built in checks. If the system fails during execution, all work will need to be redone from the beginning. The DAS development team will be adding check-pointing to the system at a later point. When check-pointing is added, the system will note which phase of the top-down algorithm executed last and it will restart execution at that point.

The system has a growing number of self-tests that are executed using the Python "py.test" framework. These tests will check both the code and the execution environment. The "py.test" will be run by the DAS prior to the start of the top-down algorithm, so that failures can be rapidly detected and diagnosed.

DAS is developing a framework for recording and alerting on out-of-memory or out-of-storage conditions.

DAS is based on Python 3.6 and will be tested with Python 3.7 when available. However, for the 2018 end-to-end test, DAS will be using Python 3.6.

DAS is based on Gurobi 7.5; version 8.0 is now available. DAS will be using version 7.5 for the end-to-end test and will move to version 8.0 in Q2 2019.

DAS assumes that the Gurobi license manager will be available. If the license manager is not available, or if a license is not available, the client code will retry until it is. A RETRY LIMIT is not currently specified.

PERFORMANCE ENGINEERING CONSIDERATIONS

DAS system performance depends on the following considerations:

- File transfer performance from S3 to EMR
- Performance of EMR distributing Python tasks
- Performance of the Java-Python gateway
- Performance of the Gurobi optimizer
- File transfer performance from DAS back to S3

PERFORMANCE METRICS

The DAS developers have developed a system for capturing the utilization of memory and CPU resources and matching them to individual runs of the DAS. The DAS developers will perform this by capturing per-process and per-CPU usage every 5 seconds using DFXML and aggregating the results on the DAS EMR MASTER node, then transferring those results to S3 storage bucket. Separately, each use of the Gurobi optimizer captivates CPU usage, CPU load, memory usage, and other process information. This is all used to monitor system performance and tune the use of Gurobi during development.

SYSTEM PLATFORM AND DESIGN

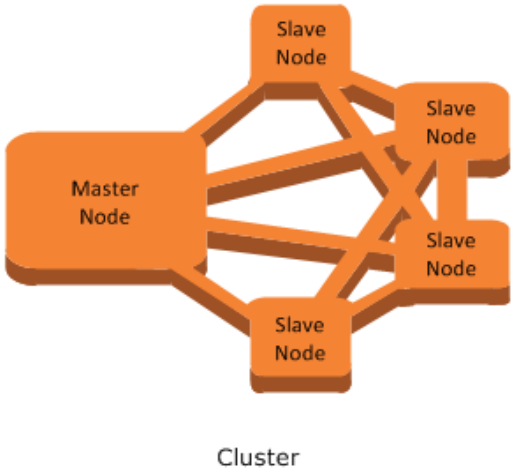
DAS utilizes Amazon EMR which is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data.

The central component of Amazon EMR is the cluster. A cluster is a collection of Amazon Elastic Compute Cloud (Amazon EC2) instances. Each instance in the cluster is called a node. Each node has a role within the cluster, referred to as the node type. Amazon EMR also installs different software components on each node type, giving each node a role in a distributed application like Apache Hadoop.

The node types in Amazon EMR are as follows:

- **Master node**: A node that manages the cluster by running software components to coordinate the distribution of data and tasks among other nodes—collectively referred to as slave nodes—for processing. The master node tracks the status of tasks and monitors the health of the cluster.
- **Core node**: A slave node with software components that run tasks and store data in the Hadoop Distributed File System (HDFS)[5] on the cluster.
- **Task node:** A slave node with software components that only run tasks. Task nodes are optional.

Figure 3: EMR Cluster Diagram



[5] DAS currently does not use HDFS.

9 API INTERFACE BUSINESS PURPOSE

The business purpose for this data exchange is to support privacy controls to microdata in the data flow from the Census Edited File (CEF) to the Microdata Detail File (MDF). The CDL platform will serve as a storage mechanism for the Disclosure Avoidance System. The Disclosure Avoidance System (DAS) to CDL interface is necessary in order for DAS to successfully complete its processing and write non-Title 13 data back to the Census Data Lake.

INTERFACE RESPONSIBILITIES

The interface includes a one directional flow of data from the Disclosure Avoidance System to the CDL. The DAS is responsible for providing the MDF to the CDL to allow for microdata tabulation. This interface will be executed using the SOA Managed File Transfer (MFT) process. The MDF file is structured as 2 tables for each of the 50 states, DC and PR.

THE 2018 E2E MDF DELIVERY CONSISTS OF TWO PIPE-DELIMITED ASCII TEXT FILES:

- MDF_UNIT.txt
- MDF_PER.txt

DISCLOSURE AVOIDANCE - CDL DATA

The MDF was delivered from the DAS to the CDL in the form of pipe-delimited ASCII text files. DAS delivered the pipe-delimited ASCII files (2 tables for each state) to S3 using Spark. These "files" will appear as directories containing multiple parts. Spark convention is to designate parts as part-0001, part-0002, etc. The program **s3cat.py** located in **das_decennial** then used an AWS S3 API to combine these files into a single file that could be moved back to the CDL.

BOUNDARY DATA STRUCTURE

The MDF will be delivered from the DAS to the CDL in the form of pipe-delimited ASCII Text files.

Table 9: Boundary Data Structure Details

Data Structure Unique Identifier	Name	Data Structure Characterization	Unique Identifier of Data Elements	Relationship (Order)	Exchange Format
MDF.Person	Microdata Detail File (MDF) Person Data	*Non Title 13*	*MDF.PersonData*	n/a	*Pipe-delimited ASCII*
MDF.Unit	Microdata Detail File (MDF) Unit Data	*Non Title 13*	*MDF.UnitData*	n/a	*Pipe-delimited ASCII*

For Boundary Data Structure see MDF Specs in Architecture section.

PERFORMANCE CONSIDERATIONS

The 2018E2E MDF delivery will consist of two pipe-delimited ASCII text files: MDF_UNIT.txt and MDF_PER.txt. The two text files will be less than 2GB in size. Files will only be transferred once unless file transfer issues occur. The DAS team will check for file transfer failures, by utilizing the SOA Trace Monitoring API.

10 DAS STANDALONE START GUIDE ON AWS

As part of this release, the capability to reproduce results in a standalone Amazon Web Services (AWS) EC2 instance was rolled into the codebase. What follows is a guide to reproduce a privacy-controlled Microdata Detail File (MDF) from 1940s IPUMS dataset outside of the Census Bureau environment.

While this section details the specifics of configuring and running the application with an EC2 instance, it can be adapted to run in another appropriately configured Linux environment. While the details of this adaptation are left to the user, this guide will provide exact steps and appropriate commands and scripts for a successful run in the specified EC2 Amazon Machine Image (AMI).

Note: The following instructions require a *Gurobi Optimizer* license, which is free for academic purposes, and can be requested as an evaluation for commercial users. Visit Gurobi for further details regarding this required license.

PROVISIONING THE AWS EC2 INSTANCE

First, acquire a system that will run DAS. This guide presumes the usage of an EC2 instance provisioned by the user in their AWS account.

To create an EC2 instance, refer to the Getting Started guide. Sign up for an AWS account here.

During the EC2 instance creation process, select the following recommended options:

On Step 1: Select Amazon Linux 2 AMI (HVM), SSD Volume Type
On Step 2: Select m5.xlarge
On Step 4: Set storage size to 200 GB
On Step 6: Create a new security group. Select "My IP" for the SSH Source

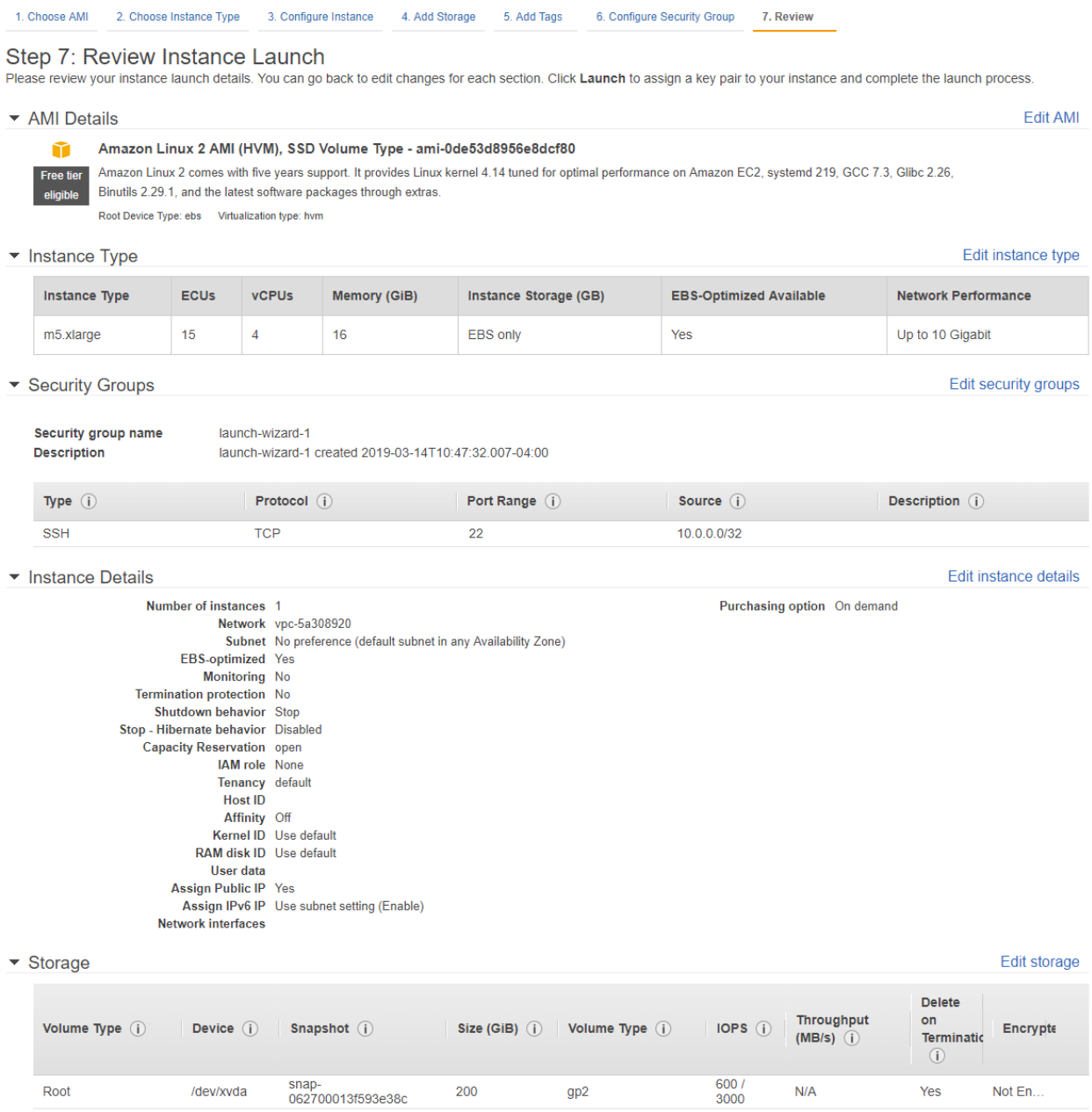
All other configurations option should remain as default. Selecting these options will result in the recommended instance, as shown in **Figure 4** on the following page.

When the Launch button is clicked, a popup will require a key pair to be selected. Create a new key pair, named ec2, and then click Download Key Pair to download the ec2.pem private key pair file. This file is necessary for the new EC2 instance via SSH and SFTP. By default, the permissions of the ec2.pem file are too open for AWS to accept. Restrict the permissions of the file with this command:

```
chmod 0400 ec2.pem
```

Note: With this configuration, DAS takes approximately 5-6 hours to run on the 1940s IPUMS dataset, which will incur a charge of less than $10 USD as long as the instance is terminated soon after results are generated.

Figure 4: Recommended AWS EC2 Configuration



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▾ AMI Details — Edit AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0de53d8956e8dcf80

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

▾ Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
m5.xlarge	15	4	16	EBS only	Yes	Up to 10 Gigabit

▾ Security Groups — Edit security groups

Security group name launch-wizard-1
Description launch-wizard-1 created 2019-03-14T10:47:32.007-04:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	10.0.0.0/32	

▾ Instance Details — Edit instance details

Number of instances 1
Network vpc-5a308920
Subnet No preference (default subnet in any Availability Zone)
EBS-optimized Yes
Monitoring No
Termination protection No
Shutdown behavior Stop
Stop - Hibernate behavior Disabled
Capacity Reservation open
IAM role None
Tenancy default
Host ID
Affinity Off
Kernel ID Use default
RAM disk ID Use default
User data
Assign Public IP Yes
Assign IPv6 IP Use subnet setting (Enable)
Network interfaces

Purchasing option On demand

▾ Storage — Edit storage

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Terminatic	Encrypte
Root	/dev/xvda	snap-062700013f593e38c	200	gp2	600 / 3000	N/A	Yes	Not En...

UPLOADING DAS AND 1940S IPUMS DATASET

This document is accompanied by an archive containing the DAS codebase, which will be uploaded to the EC2 instance. Additionally, acquire the 1940s IPUMS dataset, which is publicly available here. **Note:** Create a free account in order to download the dataset. Also, disable popup blocking if the download link doesn't work once logged in and email verified.

Note: The following instructions assume the DAS release package and 1940s IPUMS dataset are locally accessible. Alternatively, download them directly from the EC2 instance to avoid uploading them over a personal connection. That procedure, however, is not detailed here.

With SSH access configured to an external accessible EC2 instance, DAS and the dataset can be loaded onto the instance. For convenience, two methods for doing so are outlined here. The first is with the OpenSSH `sftp` CLI utility and the second is with FileZilla. However, any client supporting SFTP and private key authentication can be used.

Option 1: SFTP

With OpenSSH `sftp` already installed, place the files `das_decennial_e2e.tar.gz`, `EXT1940USCB.dat.tar.gz`, and `ec2.pem` (the access key downloaded from AWS when setting up the EC2 instance) into the current working directory. The following commands will upload the files to the locations on the EC2 instance expected by the rest of this guide:

```
$ sftp ec2-user@${YOUR_EC2_PUBLIC_IP} -i ec2.pem
sftp> put das_decennial_e2e.tar.gz /home/ec2-user
sftp> mkdir /home/ec2-user/das_files
sftp> put EXT1940USCB.dat.gz /home/ec2-user/das_files
sftp> exit
```

Option 2: FileZilla

The following is a list of instructions for connecting to the EC2 instance with FileZilla, using the access key downloaded from IAM AWS,

1. Edit (Preferences) > Settings > Connection > SFTP, Click "Add key file"
2. Browse to the location of the .pem file and select it.
3. A message box will appear asking permission to convert the file into ppk format. Click Yes, then assign the file a name and store it somewhere.
4. If the new file is shown in the list of Keyfiles, then continue to the next step. If not, then click "Add keyfile..." and select the converted file.
5. File > Site Manager Add a new site with the following parameters:

   Host: The public DNS name of the EC2 instance, or the public IP address of the server
   Protocol: SFTP
   Logon Type: Normal
   User: ec2-user
   Press Connect Button - If saving of passwords has been disabled, a prompt appears that the logon type will be changed to 'Ask for password'. Say 'OK' and when connecting, at the password prompt push 'OK' without entering a password to proceed past the dialog.

   Note: FileZilla automatically figures out which key to use. Do not specify the key after importing it as described above.

Once connected, upload the DAS archive to */home/ec2-user/* and the 1940s IPUMS dataset (*EXT1940USCB.dat.gz* by default, *EXT1940USCB.dat* uncompressed) to */home/ec2-user/das_files/* (creating directory *das_files* as appropriate).

SETTING UP FOR DAS EXECUTION

Once uploaded files are in place, connect to the EC2 instance for terminal access via SSH (refer to this guide for instructions).

The following is a short list of commands required to configure the environment for a successful DAS run on the 1940s IPUMS data (*italics* indicating shell commands).

Extract DAS archive:

```
tar xzf ~/das_decennial_e2e.tar.gz -C ~
```

Uncompress the IPUMS 1940s dataset (if not already uncompressed):

```
gunzip ~/das_files/EXT1940USCB.dat.gz
```

Run standalone prep (downloads and installs all necessary tools and libraries):

```
~/das_decennial/etc/standalone_prep.sh
```

Get Gurobi license (commercial):

```
grbgetkey ${YOUR_KEY_CODE}
```

-OR-

Get Gurobi license (academic)
Since the Gurobi license for academia is validated by checking against a list of known university addresses, establish an SSH tunnel for the license activation tool through your university – see this guide for details:

```
ssh -L8008:apps.gurobi.com:80 ${USERNAME}@${UNI_SSH_ADDR}
```

In another terminal on the EC2 instance:

```
grbgetkey --verbose --port=8008 --server=127.0.0.1 ${YOUR_KEY_CODE}
```

Optional: Update dataset location (default: */home/ec2-user/das_files/*)

```
vim ~/das_decennial/standalone_setup.sh
```

Optional: Update Gurobi license location (default: */home/ec2-user/gurobi.lic*)

```
vim ~/das_decennial/E2E_1940_STANDALONE_CONFIG.ini
```

RUNNING DAS

Once configured, running DAS in standalone mode is a simple task.

```
cd ~/das_decennial; ./run_1940_standalone.sh
```

On an EC2 instance of the recommended configuration, this operation will take roughly 5 hours and 40 minutes to complete. By default, the resultant dataset will be stored in:

```
/home/ec2-user/das_files/output
```

System resource consumption can be checked as a sign that DAS is running as expected. *htop* can be used to view core utilization and memory allocation, as well as track processes created by Spark. If the default configuration provided is used, running *htop* should show full utilization on all available cores.

```
sudo yum install htop
htop
```

Once DAS has finished running, the results folder will contain a certificate (MDF_CERTIFICATE.pdf) detailing runtime, modules used, privacy budget, and an overall error metric. The datasets themselves (*MDF_PER.dat* and *MDF_UNIT.dat*) are the concatenated version of the parts stored in their respective folders (*MDF_PER.txt* and *MDF_UNIT.txt*). *MDF_PER.dat* and *MDF_UNIT.dat* can be read as csv, most conveniently via pandas.

For example, via a python interactive terminal:

```
import pandas
p = pandas.read_csv('MDF_UNIT.dat')
```

Also, it can quickly be verified that the input and output datasets are the same size record-wise by word-count:

```
wc -w ~/das_files/output/MDF_PER.dat ~/das_files/output/MDF_UNIT.dat
```

Total should match word count of:

```
wc -w ~/das_files/EXT1940USCB.dat
```

For quick inspections purposes, an info script has been included which will print MDF headers and a specified number of records (using the `-r` flag), and can be run on a zip containing the MDF generated after a successful standalone run:

```
python3 ~/das_decennial/daszipinfo.py -r 5 ~/das_files/output/MDF_RESULTS.zip
```

It is recommended to download the contents of this folder and terminate the EC2 instance [**Warning:** Instance termination will result in loss of all data in the volume associated with the instance under the default configuration].

11 ACRONYMS

Table 10: Acronyms

Acronym	Meaning
ACSO	American Community Survey Office
AES	Advanced Encryption Standard
AIAN	American Indian or Alaska Native
AMI	Amazon Machine Image
AWS	Amazon Web Services
CED-DA	Center for Enterprise Dissemination - Disclosure Avoidance
CEF	Census Edited File. CEF processes the CUF, the tabulation geography table, and produces tables of edited data
CDL	Census Data Lake
CIS	Center for Internet Security
CUF	Census Unedited File, consisting of tables that are an input to the CEF
CMK	Customer Master Key
DAS	Disclosure Avoidance System
DCMD	Decennial Census Management Division
DITD	Decennial Information Technology Division
DSEPC	Data Stewardship Executive Policy Committee
DP	Differential Privacy
E2E	End-to-End
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
EMR	Elastic Map Reduce
GQ	Group Quarters
HDFS	Hadoop Distributed File System
IPT	Integrated Project Team
KMS	Key Management Service
MDF	Microdata Detail File
MDNH	multi-dimensional national histogram
MFT	Managed File Transfer
NHPI	Native Hawaiian or Other Pacific Islander
POP	Population Division
RPO	Response Processing Operation
TEA	Type of Enumeration Area
TI	Technical Integration
TSD	Technical Specification Document
S3	Simple Storage Service
SEHSD	Social, Economic, Housing and Statistics Division
SSE	Server-Side Encryption
SNS	Simple Notification Service
SOR	Some Other Race or Ethnicity