XSEDE
Infrastructure-as-a-Service
Use Cases

October 30, 2016

Version 2.2

This document outlines typical XSEDE user needs in the "Infrastructure-as-a-Service (IaaS)" area. IAAS-1 through IAAS-4 describe how XSEDE IaaS services are used by end users. IASS-6 through IAAS-7 describe how XSEDE IaaS services are shared within a research community with shared computing needs. IAAS-8 and IAAS-9 describes how an application or science gateway developer can automate use of XSEDE IaaS services via a software API. Table 1 shows the correspondence of these use cases to current XSEDE SP resources.

IAAS-1: Request to use an IaaS host for research

An XSEDE **researcher** or **science gateway developer** needs to request access to an XSEDE IaaS hosting service. If the request is approved, the user will be authorized to use the hosting service. We assume that the user has already registered with XSEDE.

In most cases, the **researcher** or **gateway developer** wants to experience it as follows.

1. The user logs into the XSEDE User Portal (XUP) using his/her XSEDE identity.
2. The user navigates to the XSEDE allocation section of the XUP.
3. The user follows the allocation instructions and submits an allocation request.
4. The user receives a response from the allocation system.

It's always like the steps above, except when an XSEDE service provider chooses to offer a *discretionary allocation* on their own systems. In these cases, Steps 1-3 might not occur, and Step 4 may be the first interaction the end user has with the system.

We'll accept any solution to this problem, as long as the amount of time between request submission and response in Step 4 is consistent with any expectations set in Step 3.

IAAS-2: Self-service acquire and use a virtual computer system

An **XSEDE-allocated researcher** needs exclusive access to a hosted virtual computer system that satisfies a specific set of requirements. We assume the user has previously registered with XSEDE and has been authorized to use at least one XSEDE IaaS host.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher logs into an IaaS management interface using his/her XSEDE identity.
2. The researcher constructs and submits a request for a virtual computer system. (See below for examples of system requirements.)
3. The management interface responds with a success or failure indication. In the case of success, the interface provides either: (a) a pointer to the system (IP address, DNS name, or other address appropriate for the specified system), or (b) an estimate of the time to allocate the system and a functional pointer or callback mechanism that provides access when the system is available.
4. The researcher connects to the system and conducts his/her research.
5. After completing his/her work, the researcher submits a release request. When this happens, the virtual computer system is no longer accessible and any associated physical resources are released and made available for other use.

It's always like the steps above, except when:

1. In Step 3, the management interface might redirect the researcher's request to another cloud resource within XSEDE, a campus-level resource, or a commercial provider.

2. In Step 5, the user might specify that storage objects or file systems associated with the virtual system should be retained for future use.
3. In Step 5, the user might specify that the virtual system be put into a "sleep" mode, such that it can later be quickly restored to active use. In that case, the physical resources that support the system can be freed, but the system state should be preserved in storage.

We'll accept any solution to this problem, as long as in Step 2, the management interface allows the researcher to specify any of the following that are appropriate for the host system(s): number of nodes, number of cores per node, type of cores, network interconnect type, network configuration, memory size, storage size(s), operating system, preinstalled software.

IAAS-3: Self-service acquire and use a hosted storage volume

An **XSEDE-allocated researcher** needs exclusive access to a hosted storage volume that satisfies a specific set of requirements. We assume the researcher has previously registered with XSEDE and has been authorized to use at least one XSEDE IaaS host.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher logs into an IaaS management interface using his/her XSEDE identity.
2. The user constructs and submits a request for a storage volume. (See below for examples of system requirements.)
3. The management interface responds with a success or failure indication. In the case of success, the interface provides either: (a) a pointer to the system (IP address, DNS name, or other address appropriate for the specified system), or (b) an estimate of the time to allocate the system and a functional pointer or callback mechanism that provides access when the system is available.
4. The user connects to the storage volume and reads and writes data to and from the volume to conduct his/her research.
5. After completing his/her work, the user submits a release request. When this happens, the storage volume is no longer accessible and any associated physical resources are released and made available for other use.

It's always like the steps above, except when:

1. In Step 3, the management interface might redirect the researcher's request to another cloud resource within XSEDE, a campus-level resource, or a commercial provider.
2. In Step 5, the service provider may offer the option of having the contents of the storage volume retained in archival storage for potential reuse in the future.

We'll accept any solution to this problem, as long as in Step 2, the management interface allows the researcher to specify any of the following that are appropriate for the host system(s): storage size, storage attributes (e.g., performance, reliability), network interconnect type and/or configuration, backup policy, access protocols (e.g., FTP, NFS, local mount on a virtual compute system).

IAAS-4: Request and use a virtual cluster

An **XSEDE-allocated researcher** obtains exclusive access to a virtual cluster (a predefined suite of hosted computer systems and/or storage volumes) <u>with expert assistance by a service provider</u>. We assume the researcher has previously registered with XSEDE and has been authorized to use at least one XSEDE IaaS host.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher composes a description of the desired suite of hosted computer systems and/or storage volumes and delivers this description to the service provider of an IaaS host system.
2. The service provider responds to the request with a set of pointers to the virtual cluster and its constituent elements.
3. The researcher uses the virtual cluster for his/her research.
4. After completing his/her work, the researcher submits a release request. When this happens, the virtual cluster is no longer accessible and any associated physical resources are released and made available for other uses.

It's always like the steps above, except that in Step 4 the service provider may offer the option of having system images, storage volumes, or other filesystems retained in archival storage for potential reuse in the future. Also, in Step 2, the service provider might need to seek clarification from the end user about his/her request before returning the set of pointers to the requested resources.

We'll accept any solution to this problem, as long as in Step 1, the management interface allows the researcher to specify any of the following that are appropriate for the host system(s): number of nodes, number of cores per node, type of cores, memory size, operating system, preinstalled software, storage size, storage attributes (e.g., performance, reliability), backup policy, access protocols (e.g., FTP, NFS, local mount on a virtual compute system), <u>network interconnect type and/or configuration</u>.

IAAS-5: Publish a virtual computer system or storage volume

An **XSEDE-allocated researcher** needs to "publish" a hosted computer system or hosted storage volume that may be of interest to his/her research community. Once published, other members of the XSEDE community can recreate the system or volume for their own use. We assume that the researcher already has an XSEDE identity, currently has an allocation to use at least one XSEDE IaaS service provider, and currently has a virtual computer system or storage volume allocated for his/her use on an IaaS hosting service.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher prepares his/her virtual computer system or storage volume for publishing, putting it into a state that will be useful to other researchers who replicate the system or volume. (This typically means installing and configuring software, loading data, and cleaning things up to remove unnecessary detail. It might include adding documentation files describing how to use the system or volume.)

2. The researcher submits a request to the system to publish his/her virtual computer system or storage volume. The request could be via a Web interface or an email message, and it includes an identifier for the system or volume and metadata that describes the purpose in terms that will be helpful to other researchers. It might also include a preferred repository for the published system if the service provider supports more than one.
3. The IaaS service provider prepares the system or volume for publishing and submits it to the repository for publication.
4. The IaaS service provider notifies the researcher that his/her system or volume is being published and provides an identifier or pointer that allows the researcher to locate the submission in the repository and follow it through the publication process.

It's always like the steps above, except that some host systems might not require Step 3 (preparation for publication by the service provider). In this case, the researcher's request in Step 2 includes the submission to the repository, eliminating Step 3, and the notification in Step 4 comes from the repository rather than from the IaaS service provider.

We'll accept any solution to this problem, as long as the pointer given to the researcher in Step 4 allows the researcher to obtain an identifier that he/she can share with research colleagues once publication is complete.

IAAS-6: Discover a relevant computer system or storage volume

A **researcher** needs to discover specific virtual computer systems or storage volumes that might be relevant to their research. The researcher may or may not have registered with XSEDE and may or may not have an active allocation to use XSEDE services.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher accesses the search interface for a repository of previously published virtual computer systems and/or volumes.
2. The researcher uses the interface to execute a search for relevant research topics, keywords, or other distinguishing identifiers.
3. The researcher obtains identifiers from the repository interface that can be used to retrieve specific virtual computer systems or storage volumes.

We'll accept any solution to this problem, as long as the identifier(s) returned from the repository in Step 3 are sufficient to serve as input for use case IAAS-7.

IAAS-7: Create an instance of a published computer system or storage volume

An **XSEDE-allocated researcher** needs exclusive access to a new instance of a computer system or storage volume that was previously published by another researcher. We assume the end user has previously registered with XSEDE, is already authorized to use one or more IaaS hosts, and has an identifier for a previously published virtual computer system or storage volume.

In most cases, the **researcher** wants to experience it as follows.

1. The researcher logs into an IaaS management interface using his/her XSEDE identity.
2. The researcher submits a request to create a new instance of a published virtual computer system or storage volume, including the identifier for the published item. The new instance may be requested via human interface (see IAAS-2 and IAAS-3) or via an API (see IAAS-8).
3. The management interface responds with a success or failure indication. In the case of success, the interface provides either: (a) a pointer to the system (IP address, DNS name, or other address appropriate for the specified system), or (b) an estimate of the time to allocate the system and a functional pointer or callback mechanism that provides access when the system is available.
4. The researcher connects to the system or volume using the pointer provided in Step 3 and conducts his/her research.
5. After completing his/her work, the researcher submits a release request. When this happens, the virtual system or volume is no longer accessible and any associated physical resources are released and made available for other uses.

It's always like the steps above, except when:

1. In Step 3, the management interface might redirect the end user's request to another cloud resource within XSEDE, a campus-level resource, or a commercial provider.
2. In Step 5, the user may specify that storage objects or file systems associated with the virtual system or storage volume should be retained for future use.

We'll accept any solution to this problem as long as in Step 2, the management interface allows the researcher to specify any of the following that are appropriate for the host system(s): number of nodes, number of cores per node, type of cores, memory size, storage size(s), storage attributes (e.g., performance, reliability), network interconnect type and/or configuration, backup policy, access protocols (e.g., FTP, NFS, local mount on a virtual compute system).

IAAS-8: Dynamically acquire and manage hosted computer systems and/or storage volumes

An **XSEDE science gateway developer** needs to develop an application that can automatically acquire and manage hosted computer systems and/or storage volumes via an Application Programmer Interface (API). We assume that the gateway developer has previously registered with XSEDE and has been authorized to use at least one XSEDE IaaS host. Further, it is assumed that the gateway developer explained his/her need for API access in his/her authorization request and that API access was approved.

In most cases, the **gateway developer** wants to experience it as follows.

1. The gateway developer requests a service endpoint and client credentials for the IaaS management API, referencing a specific XSEDE project and/or allocation, and receives these items from the IaaS host's service provider.

2. Using the service endpoint, client credentials, and API documentation provided by the IaaS host's service provider, the gateway developer codes his/her application to use the API.
3. The gateway developer deploys his/her application and it is used by its own community of end users. While in operation, the application makes API requests for allocating, managing, and releasing, hosted computer systems and/or storage volumes. Each request is handled in a manner consistent with the API documentation.
4. When the gateway developer is ready to conclude the gateway application's deployment (because the scientific objective has been met or because his/her XSEDE allocation has expired), the IaaS host's service provider invalidates the developer's client credential and releases any hosted resources that were still in use by the application. Subsequent API requests from the gateway application are rejected by the IaaS host.

It's always like the steps above, except that the IaaS host's service provider may temporarily invalidate the gateway's client credential at any time to prevent the gateway from allocating, managing, or releasing any resources. Reasons for doing this might include a security incident or a planned maintenance period.

We'll accept any solution to this problem as long as:

1. In Step 2, API documentation should be readily available to developers, including those who don't currently have access credentials. The documentation should allow client coding with a moderate degree of expertise. The documentation should include coding examples for commonly used functions.
2. In Step 3, the IaaS management API should provide immediate responses to all requests. (Maximum round-trip times should be measured in single-digit seconds.) If a request can't be fulfilled or rejected immediately, an acknowledgement of the request should be provided immediately with a callback identifier that can be used to check on the status of the request.
3. In Step 4, determination and communication of the "ready to conclude" status of a gateway has been left intentionally vague. It is a joint determination by the service provider and the gateway developer and it may occur for any of several reasons. The service provider and XSEDE are jointly responsible for ensuring that the expected duration of use of the hosted services is clear to the gateway developer.

IAAS-9: Reliably acquire and manage hosted computer systems and/or storage volumes

This use case is identical to IAAS-8 with the following addition. *(Although this additional requirement is easy to describe, we recognize that a solution that satisfies it may be significantly more difficult to provide than one that doesn't.)*

We'll accept any solution to this problem as long as the API provided by XSEDE automatically resolves common issues and only returns control to the calling application when the request succeeds or cannot

reasonably be satisfied without human intervention. (To the greatest degree possible, the calling application should not have to handle common exceptions.)

IAAS-10: Add a science gateway operator to an API-enabled IaaS allocation

An **XSEDE-allocated researcher** needs to be able to add an XSEDE science gateway operator to his/her API-enabled allocation on an IaaS host. This will allow the researcher to use the gateway as a means for using his/her allocation. (When the researcher uses the gateway, the gateway will use the researcher's allocation.) We assume that the researcher has previously registered with XSEDE and has been authorized to use at least one XSEDE IaaS host. We also assume that the researcher explained his/her plan to use the gateway in his/her authorization request and that API access was approved. Finally, we assume that the gateway is already set up for use case IAAS-8 or use case IAAS-9 and has the ability to use separate client IDs and credentials for individual gateway end users.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher logs into the XSEDE User Portal (XUP) using his/her XSEDE identity.
2. Then, the researcher navigates to the "Add User" section of the XUP.
3. Then, the researcher selects the project corresponding to the IaaS allocation and adds the gateway developer to the project.
4. Finally, the XSEDE user who was added to the project receives a notification that he/she has been added to the project, including the project identifier, which should in turn allow the new project member to use the researcher's allocation via use case IAAS-8 or IAAS-9.

We'll accept any solution to this problem as long as the XSEDE-allocated researcher can disable the new project member's access to the allocation at a later time, after which the (former) project member will no longer be able to use the allocation with any IAAS use case.

Glossary

Computer system image - A stored, static representation of the state of a virtual computer system that can be restored as a running virtual system

Infrastructure-as-a-Service - aka, "**IaaS**;"a form of cloud computing that provides virtualized computing resources over the Internet

IaaS host system - a system that offers IaaS to end users

IaaS management interface - any interface (user or programmer) that allows end users to manage their use of an IaaS host system; the interface may be web-based, command-line, or an API

Instance - a distinct copy of a virtual computing resource, either a computer system or a storage volume. E.g., if a single computer system image is used to initialize multiple virtual computer systems, then each virtual system is an *instance* of the original image.

Science gateway - a research or science application that allows a large community of researchers, scientists, and/or students to use a set of broadly useful tools, running or stored on XSEDE resources, without all members of the community having to obtain individual XSEDE allocations

Storage volume - a logical file system that can be "attached" to a computer system (virtual or physical) in order to access and/or manipulate the contents of the file system

Virtual cluster - a suite of virtual computing resources and/or storage volumes managed and used together for a single research purpose

Virtual computer system - a logical set of computing resources used by an end user for a specific purpose and implemented as a set of VMs, storage volumes, and virtual network interconnect

Virtual machine - aka, "**VM**;" a logical computer that is implemented by software running on a host system; the software provides the appearance and functionality of a physical system

XSEDE System, the - the set of services that are offered by the XSEDE community to researchers and scientists and the physical resources that implement those services

Correspondence to current XSEDE resources

At the present time, these use cases appear in the user documentation for several XSEDE resources that have IaaS features. (See Table 1.)

Use case	Bridges[1]	Comet[2]	Jetstream[3]
IAAS-1	X	X	X
IAAS-2		X	X
IAAS-3			X
IAAS-4	X		
IAAS-5			X
IAAS-6			X
IAAS-7			X
IAAS-8		X	X
IAAS-9			
IAAS-10	X	X	X

Table 1. Current XSEDE resources that support each use case in this document.

[1] Bridges User Guide. Retrieved August 15, 2016. (https://www.psc.edu/index.php/bridges/users/user-guide)
[2] Comet User Guide. Retrieved August 15, 2016. (http://www.sdsc.edu/support/user_guides/comet.html)
[3] Jetstream Public Wiki. Retrieved August 15, 2016. (http://wiki.jetstream-cloud.org/)

History

	Version	Date	Changes	Author
First use case draft	1.0	09/09/2012	Document created	Hossain, A
Edited based on feedback	1.1	10/05/2012	List of 4 use cases; glossary; draft development of first 2 use cases	Hossain, A
New version based on Bridges, Comet, & Jetstream resource deployments	2.0	08/15/2016	New XSEDE2 format, reformulated use cases; original use case from v1.1 now appears as IAAS-2.	Liming, L
Same as 2.0 but in a new XSEDE2 RACD format	2.1	09/07/2016	New XSEDE2 RACD proposed format	Liming, L
New use cases	2.2	10/30/2016	Additional use cases 9 and 10 from discussion with SGW developer Emre Brookes; further simplified descriptions; cleaned up formatting; renumbered use cases	Liming, L