DRM4G Use Cases

Table of Contents

Implemented features

Use case: Creating cloud credentials	
Description	Creates the proxy certificate that will be used to access the Federated Cloud's services
Primary Actor	User
Preconditions	• The DRM4G must be running on a terminal • The user must have edited correctly the configuration file • The user must have a valid grid certificate
Basic Flow	1. The user executes the command `drm4g id <resource_name> init`
Postconditions	• A proxy valid for 7 days will have been generated
Alternate Flow	1. The user uses the option `--lifetime` to create an identity for a specific period of time

Use case: Creating VMs	
Description	An active and accessible Virtual Machine (VM) will be created. It will be considered a *host* to whom the user can submit jobs to have executed
Primary Actor	User
Preconditions	• The DRM4G must be running on a terminal • The user must have edited correctly the configuration file • The user must have already created his x509 certificate proxy
Basic Flow	1. The user executes the command `drm4g resource create` 2. The program will read the configuration file 3. For every resource defined to create VMs, it will create as many VMs as nodes were specified
Postconditions	• As many VMs as nodes specified in the configuration file will have been created • The VM's public IP direction will be shown • Submitted jobs will also be sent to the VM
Alternate Flow	1. If for some reason there's a problem creating a VM, an error message is displayed 2. The program will continue creating the rest of the specified VMs

Use case: Adding VMs	
Description	Adds more VMs to those previously created
Primary Actor	User
Preconditions	• User must have previously created VMs
Basic Flow	1. The user executes the command `drm4g resource create`
Postconditions	• As many VMs as nodes specified in the configuration file will have been added • The VM's public IP direction will be shown • Submitted jobs will also be sent to the VM
Alternate Flow	1. If for some reason there's a problem creating a VM, an error message is displayed 2. The program will continue creating the rest of the specified VMs

Use case: Listing VMs	
Description	A list of all the accessible VMs will be displayed
Primary Actor	User
Preconditions	• User must have previously created VMs
Basic Flow	1. User runs the commands `drm4g host list`
Postconditions	• A list with information from every host will be shown, including from all of the active VMs created
Alternate Flow	

Use case: Listing resources	
Description	A list with every defined resource will be displayed. That includes the ones defined by the user and the ones created by the program for every VM created
Primary Actor	User
Preconditions	• The DRM4G must be running on a terminal
Basic Flow	1. The user runs the command `drm4g resource list --all`
Postconditions	• A list with information from every resource will be displayed
Alternate Flow	1. If there aren't any defined resources, none will appear

Use case: Sending jobs	
Description	Jobs will be sent to all available hosts to be executed
Primary Actor	User
Preconditions	• The DRM4G must be running on a terminal • The user must have edited correctly the configuration file • The user must have edited correctly a job template
Basic Flow	1. The user runs the command `drm4g job submit path/to/job/template.job`
Postconditions	• The job is executed by one of the available host. This includes those defined in the configuration file with the purpose of executing jobs and VMs that may have been previously created
Alternate Flow	1. The user uses the option `--ntasks` to determine how many times he wants to have the same job executed 2. The user uses the option `--dep` to define the job dependency list of the job

Use case: Destroying all VMs	
Description	Regardless whether they are being used o not, every VM created by the DRM4G will be destroyed
Primary Actor	User
Preconditions	• The user must have previously created VMs
Basic Flow	1. The user runs the command `drm4g resource destroy` 2. The program will eliminate all created VMs one by one
Postconditions	• There won't be any VM left
Alternate Flow	1. If for some reason there's a problem destroying a VM, an error message is displayed 2. The program will continue destroying the rest of the VMs

Forthcoming features

Use case: Listing VMs with IDs	
Description	When running the command `drm4g resource list --all`, the program will detect if the resource it's listing is a VM and list its *ID*
Primary Actor	User
Preconditions	• The user must have previously created VMs
Basic Flow	1. The user runs the command `drm4g resource list --all` 2. If the program detects that the resource it's listing is a VM, it will add to the rest of its information it displays its *VM ID*.
Postconditions	• A list with information from every resource will be displayed • VMs' information will display its ID
Alternate Flow	

Use case: Creating VMs dynamically	
Description	The user will indicate in the "resources.conf" with the configuration keys "***max_nodes***" and "***min_nodes***" how many VMs he wants created
Primary Actor	User
Preconditions	• There must be a resource in the configuration defined with the purpose for the creation of VMs • The user must have already created his x509 certificate proxy

Basic Flow	1. The program checks how may VMs are already available
	  1. If the total number of created VMs is the same as "***min_nodes***" it does nothing
	  2. If the total number of created VMs is less than "***min_nodes***" it creates as many VMs necessary to reach "***min_nodes***"
	  After "***min_nodes***" VMs have been created, if there are still a certain number of jobs in a *pending* state for a hard-coded period of time, an additional VM will be created
	  1. This will go on for every cycle until either the number of VMs has reached "***max_nodes***" or until the number of *pending* jobs goes below the aforementioned limit
	3. If the number of *pending* jobs is below the aforementioned limit but the amount of time they have been *pending* is larger than several times the aforementioned limit, an additional VM will be created as long as the number of created VMs is smaller than "***max_nodes***"

Postconditions	• As long as the billing hasn't reached it's limit, there will always be a VM available for use

Alternate Flow	If the user hasn't specified "***max_nodes***" or "***min_nodes***"
	  1. "***min_nodes***" will be given a value of zero
	  2. "***max_nodes***" won't be given a limit
	2. If only "***min_nodes***" has been specified, "***max_nodes***" will be given the same value
	3. If only "***max_nodes***" has been specified, "***min_nodes***" will be given a value of zero