### Use Case – Configure XMPP Server.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. A host or virtual machine.
  + The actor is logged into the XMPP server and has super user level privileges on that account.
* Description:
  + The use case begins when the actor runs the xmpp\_setup.py script on the command line of the dedicated XMPP server.
    - The actor must specify the username and password of the first admin user to be created by the script.
      * --user <user name>
      * --password <password>
    - The system will download and install the appropriate packages and then alter the configuration files required to successfully install the XMPP server.
    - The system shall create the admin user with the specified username and password.
    - The use case ends when the system returns a successful message for all requirements.
* Post Conditions:
  + The XMPP Server is setup and running.
  + The Administrator User is created on the XMPP Server.
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The system fails to download the required packages.
* The system does not recognize the arguments passed by the user.
* The system fails to properly install the packages required.

Related Use Cases:

* Configure VPN.

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/13/2014

Date last modified: 10/13/2014

### Use Case – Configure VPN Node.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. A host or virtual machine.
  + The XMPP server is running successfully
  + The actor is logged into the VM to be configured as the node.
  + The VPN node is accessible to the XMPP server vice versa.
* Description:
  + The use case begins when the actor runs the vpn\_node\_setup.py script on the command line of the VM.
    - The actor must specify the username and password of the user to be created by the script.
      * --xmpp-user <user name>
      * --xmpp-password <password>
    - The actor must specify the address of theVPN node, the network mask, the IP address of the XMPP server.
      * --ipop-address <ip address assigned to this VM on the VPN>
      * --ipop-mask <network mask>
      * --xmpp-host <IP address of the XMPP server>
    - The system will download and install IPOP then create the configuration file with the given arguments.
    - The system shall create the XMPP user with the specified username and password, via RPC call to the XMPP server.
    - The use case ends when the system returns a successful message for all requirements.
* Post Conditions:
  + The required configuration files have been created on the VM.
  + The User is created on the XMPP Server.
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The system fails to download the required packages.
* The system does not recognize the arguments passed by the user.
* The system fails to properly install the packages required.
* The system fails to write the configuration file.

Related Use Cases:

* Setup XMPP server.
* Join VPN
* Leave VPN

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/13/2014

Date last modified: 10/13/2014

### Use Case – Join VPN.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. A host or virtual machine.
  + The XMPP server is running successfully
  + The actor is logged into the VM to be joined into the VPN.
  + The VPN node is accessible to the XMPP server vice versa.
  + There is a correct configuration file available to the function.
* Description:
  + The use case begins when the actor runs the join\_vpn.py script on the command line of the VM.
    - The actor must specify the configuration file.
      * --conf <configuration file>
    - The system will start the ipop-tincan program with the MCCVPN controller.
    - The use case ends when the system returns a successful message for all requirements.
* Post Conditions:
  + There exists a new network interface with the IP and Mask specified in the configuration file.
  + The VM can reach other VMs through the VPN
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The system does not recognize the arguments passed by the user.
* The system is unable to read or find the specified configuration file.
* The system fails to start the ipop-tincan or MCCVPN controller.

Related Use Cases:

* Setup XMPP server.
* Leave VPN

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/13/2014

Date last modified: 10/13/2014

### Use Case – Leave VPN.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. A host or virtual machine.
  + The XMPP server is running successfully
  + The actor is logged into the VM to be removed from the VPN.
  + The VM is actually joined to the VPN
  + The VPN node is accessible to the XMPP server vice versa.
* Description:
  + The use case begins when the actor runs the leave\_vpn.py script on the command line of the VM.
    - The system will stop the ipop-tincan program and the MCCVPN controller.
    - The use case ends when the system returns a successful message for the removal of the VM from the VPN network.
* Post Conditions:
  + The IPOP network interface no longer exists on the system.
  + The VM cannot reach other VMs through the VPN
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The system does not recognize the arguments passed by the user.
* The system fails to stop the ipop-tincan or MCCVPN controller.

Related Use Cases:

* Setup XMPP server.
* Join VPN

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/13/2014

Date last modified: 10/13/2014

### Use Case – Start XMPP Server.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. Installed on a host or virtual machine.
  + The actor is logged into the XMPP server and has super user level privileges on that account.
  + A correct and complete configuration file is located on the server that is to function as the XMPP server.
* Description:
  + The use case begins when the actor runs the xmpp\_controller.py script on the command line of the dedicated XMPP server.
    - The actor must specify the usage parameter accepted by the script.
      * --start
    - The system will start the XMPP server following the parameters included within the configuration file.
    - The use case ends when the system returns a successful message for all requirements.
* Post Conditions:
  + The XMPP Server is started.
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The configuration file either incorrect or incomplete or missing.
* The system does not recognize the arguments passed by the user.

Related Use Cases:

* Configure VPN.

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/15/2014

Date last modified: 10/15/2014

### Use Case – Stop XMPP Server.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. Installed on a host or virtual machine.
  + The actor is logged into the XMPP server and has super user level privileges on that account.
  + The XMPP Service is running.
* Description:
  + The use case begins when the actor runs the xmpp\_controller.py script on the command line of the dedicated XMPP server.
    - The actor must specify the usage parameter accepted by the script.
      * --stop
    - The system shall store the state of all connections, virtual machines, paths, and VPN groups in storage.
    - The system will stop the XMPP server.
    - The use case ends when the system returns a successful message indicating that the server has been stopped successfully.
* Post Conditions:
  + The state of the objects of the system are written to disk.
  + The XMPP Server is stopped.
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The XMPP Server is not running.
* The XMPP Server is unable to connect to the database.
* The system does not recognize the arguments passed by the user.

Related Use Cases:

* Configure VPN.

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/15/2014

Date last modified: 10/15/2014

### Use Case – Restart XMPP Server.

Details:

* Actor: Administrator
* Preconditions:
  + A dedicated XMPP server with access to the internet. Installed on a host or virtual machine.
  + The actor is logged into the XMPP server and has super user level privileges on that account.
  + The XMPP Service is running.
* Description:
  + The use case begins when the actor runs the xmpp\_controller.py script on the command line of the dedicated XMPP server.
    - The actor must specify the usage parameter accepted by the script.
      * --restart
    - The system shall store the state of all connections, virtual machines, paths, and VPN groups in storage.
    - The system will stop the XMPP server.
    - The system will start the XMPP server.
    - The use case ends when the system returns a successful message indicating that the server has been stopped successfully.
* Post Conditions:
  + The last state of objects within the system are written to disk.
  + The XMPP Server is started.
  + The system returns a success message.

Alternative Courses of Action:

* None.

Exceptions:

* The XMPP Server is not running.
* The XMPP Server is unable to connect to the database.
* The system does not recognize the arguments passed by the user.

Related Use Cases:

* Configure VPN.

Decision Support

Criticality: High. This service is critical for the execution and implementation of the system. Without this service the VPN network is not feasible.

Risk: Low. The required packages are open-source and easily downloaded and installed. Documentation and configuration examples are readily available.

Constraints:

* Usability
  + The system shall provide a help menu for each available option that includes a description and example.
* Performance
  + The system shall complete the setup in less than one minute. Not including the time it takes to download the required packages as this is highly variable.
* Supportability
  + Any POSIX-compliant OS will be able to support this function.
* Implementation
  + This functionality will be implemented in its own setup script which will run in the command line. It will be written in Python. This functionality maybe called up by other functions.

Owner: Francois D’Ugard

Initiation date: 10/15/2014

Date last modified: 10/15/2014