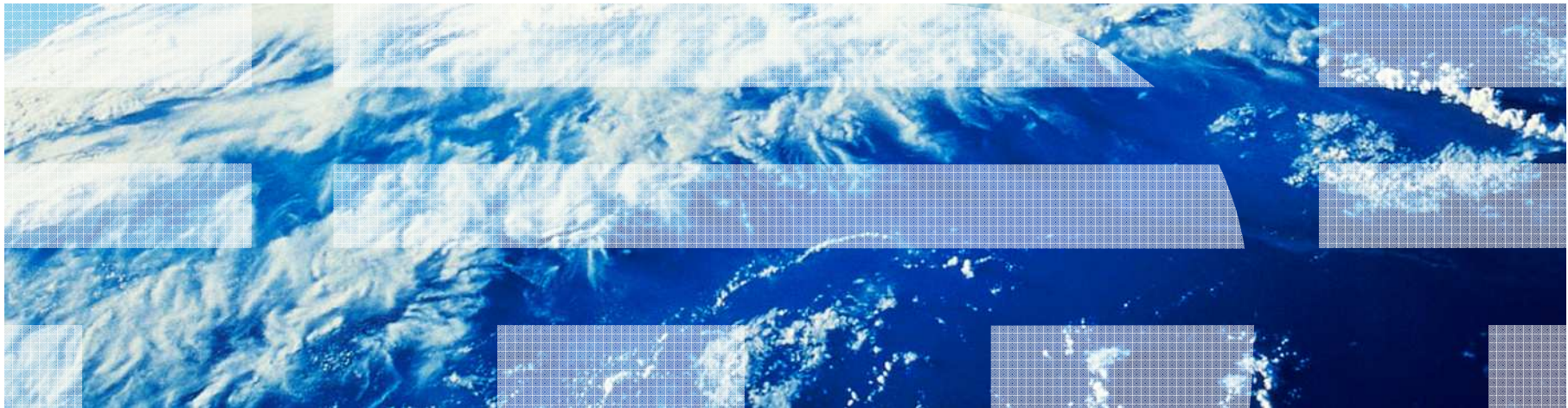


---

# COMS E6998-6 Cloud Computing and Data Center Networking

**Sambit Sahu**

<http://www.cs.columbia.edu/~sambits/>



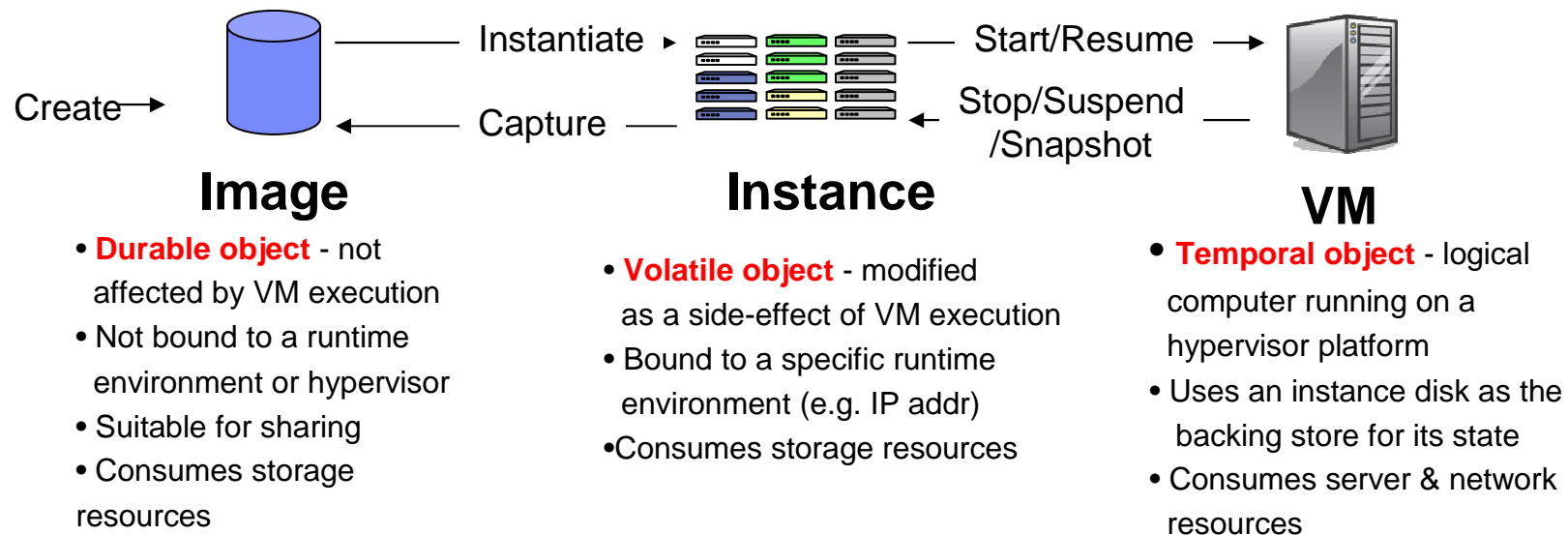
---

## This Week

- Art of Provisioning in the Cloud
  - Overview of provisioning
    - Image-based provisioning
    - Resource manager
    - Image Library
  - Composite Appliances
  - Activation process
    - Single VM
    - Composite VM

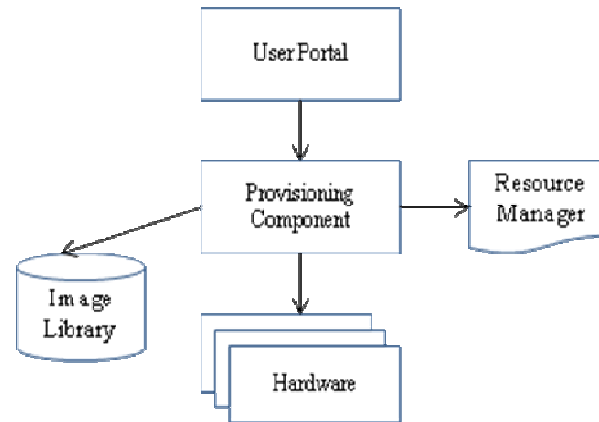
---

## Image based approach for provisioning



---

## Art of Provisioning



The Cloud resources are made available to users through a User Portal or Web Services APIs.

1. User requests are forwarded to a Provisioning Component
2. Resource Manager locates a hardware resource that has available capacity
3. Copies the image for the virtual machine from the Image Library to the target hardware resource.
4. Configures and creates the virtual machine
5. Notifies the user upon completion of the request

---

## Resource Management

- Resource Manager manages the hardware and network resources in the system.
- Resources are organized in groups and are placed in different resource pools.
  - e.g. the available host machines with information about their disk storages, memories and CPUs are placed in a system-wide host-pool, while the list of free hostnames and ipaddresses for guest virtual machine instances are specified in a defined free-guest-pool.
  - The pools are tagged allowing the VM requirements to be matched with capabilities of hosts within the pool.
- Resource Manager provides
  - Free ipaddress from the free-guest-pool
  - A host machine from the host-pool

---

## Image Library

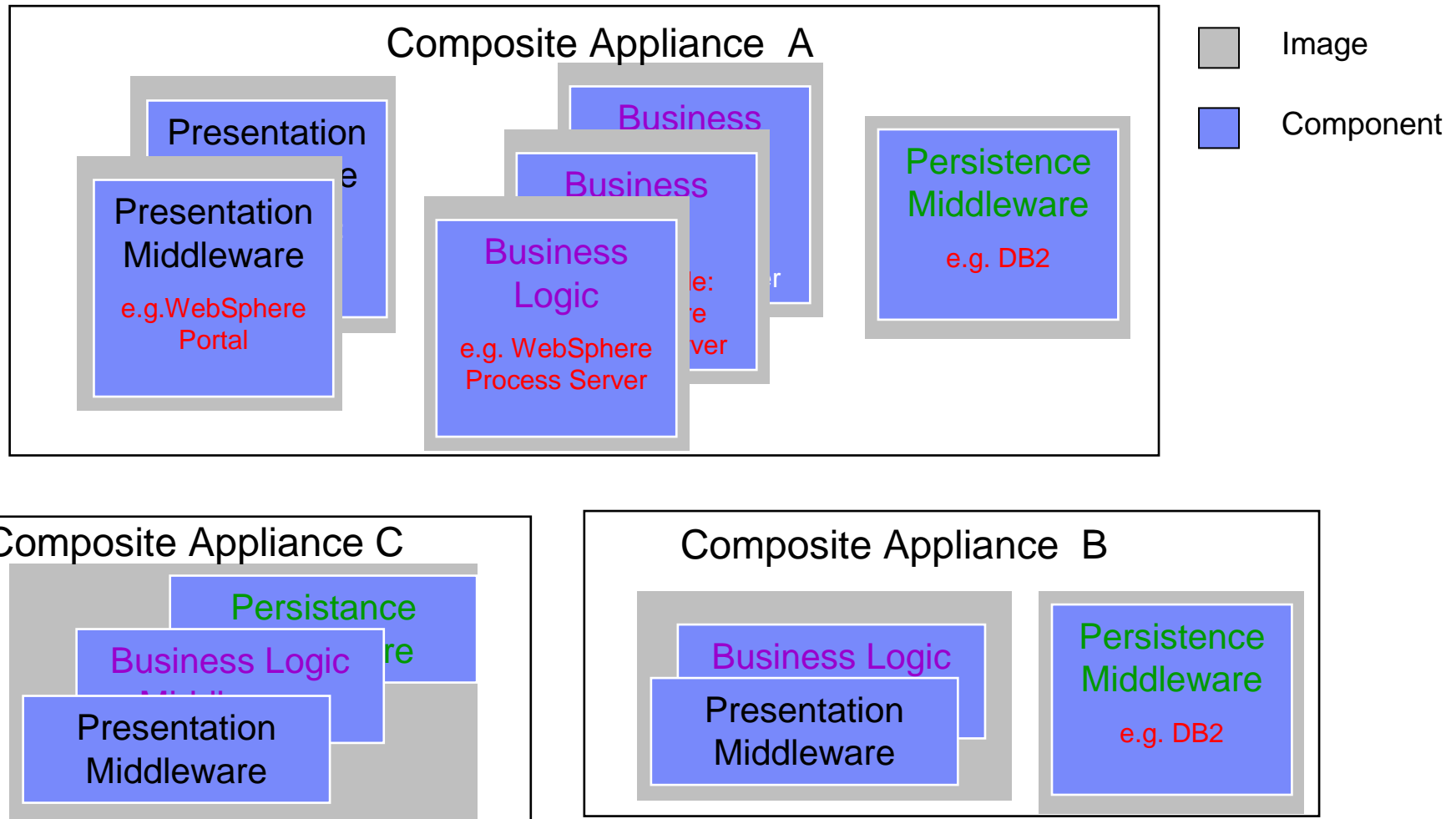
- Image Library serves as a repository for images
  
- Options
  - Standard file system
  - Using a content addressable store
    - 90% of images are the same
    - Treats images as structured data, not machines
    - Index file systems, not just images
    - Perform IT operations on views of images
    - Exploit redundancy among large numbers of images

---

## Using images as unit of deployment.

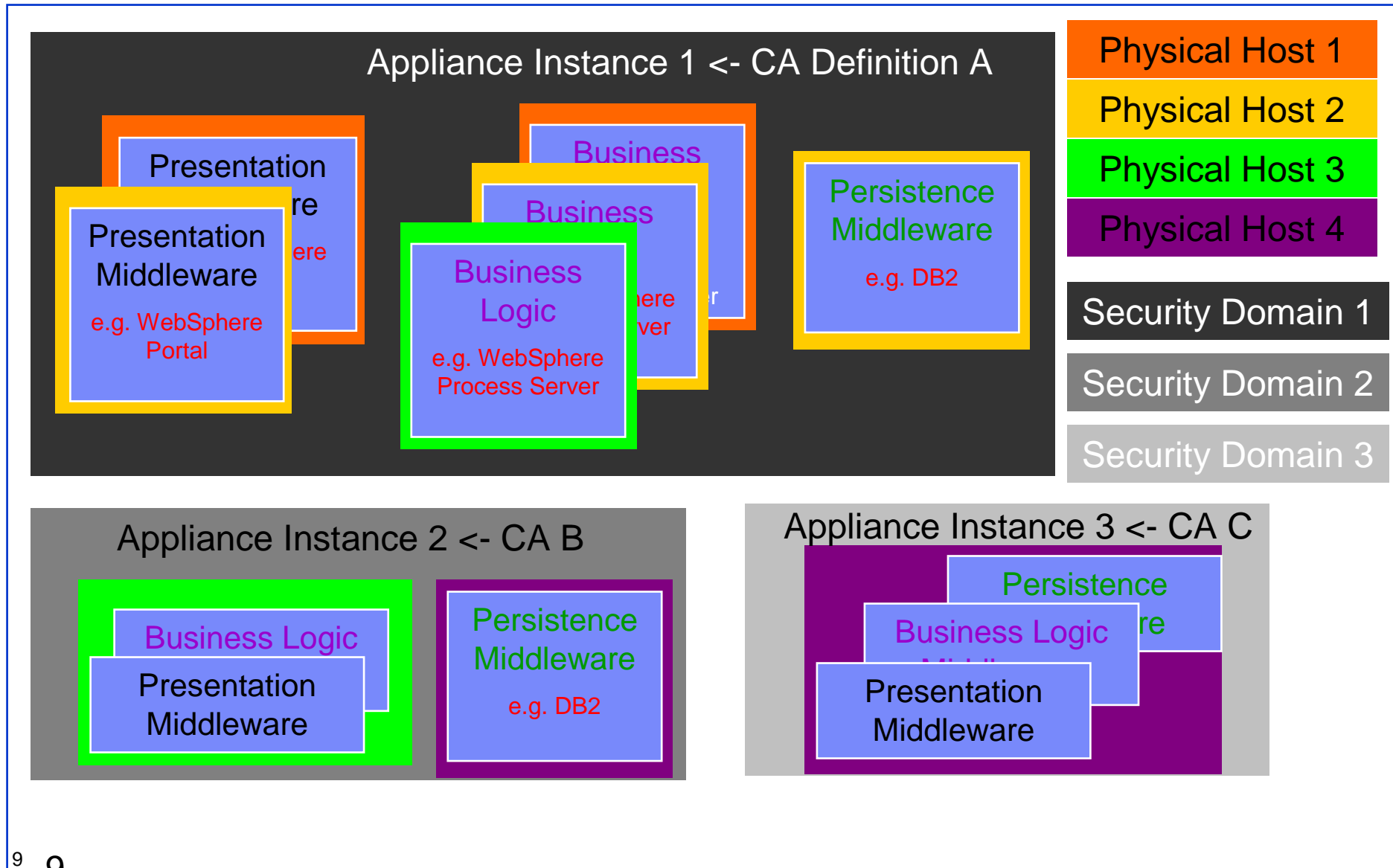
- A Composite image consists of two parts:
  - Image Definition
    - The xml definition consists of the description and set of parameters that provide additional details of the images. The xml definition is in OVF format
  - Image content
    - Tarballs of file system images for the Virtual Machines or Content Addressable Store
    - The configuration files consist of the floppy.flp/vaap file for activation engine
    - Custom configuration scripts for custom deployment type

## Example of three Composite Appliance Template definitions



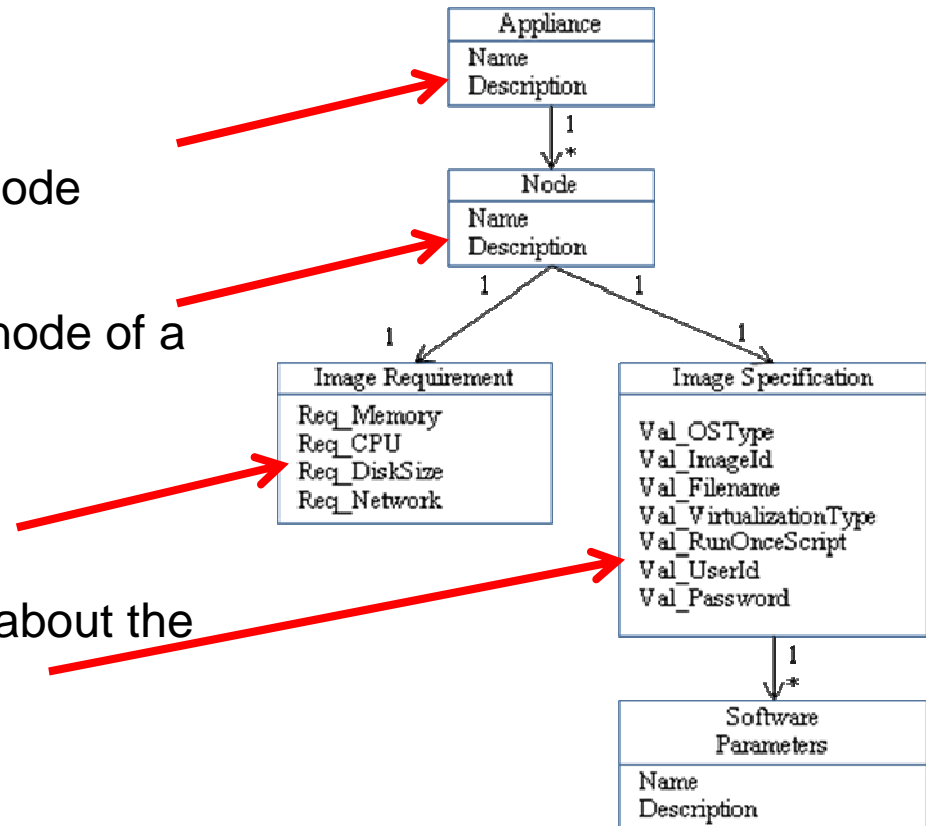


## Example of three Composite Appliance Instances



## Composite Appliance Specification

- Represented as an XML
- *Appliance* section consists of 1 or more Node sections
- *Node* section corresponds to one logical node of a solution
- *Image Requirements* specifies hardware requirements
- *Image Specification* contains information about the image and name-value pairs



---

## An example of a WebSphere Portal node XML definition

```
<node>
  <name> WebSphere Portal Server </name>
  <description> WPS for tier1 </description>
  <image-requirement>
    <req_memory> 8GB </req_memory>
    <req-cpu> 2 </req-cpu>
    <req-disksize> 40GB </req-disksize>
    <req-network> 1GB </req-network>
  </image-requirement>
  <image-specification>
    <ostype>Redhat 5u3 </ostype>
    <imageid>img-1000</imageid>
    <filename>WPimage.tgz</filename>
    <virtualization-type> KVM </virtualization-type>
    <runoncescript>"/root/fixup.sh $hostname $domain;/opt/ibm/WebSphere/AppServer/setWAS.sh
$HOST1;/opt/ibm/WebSphere/AppServer/setDB2Server.sh $HOST2;"</runoncescript>
    <userid> root </userid>
    <password> passw0rd </password>
  <software-attributes>
    <software-attribute name="userid" value="admin"/>
    <software-attribute name="passwd" value="admin"/>
    .....
    <software-attribute name="port" value="9080"/>
  </software-attributes>
</image-specification>
</node>
```

---

## Linking Image specification into a Solution

- Solution developers specify and link the components using the XML specification
  - Hostname and ipaddress is assigned via late-binding
  - Post-install configuration steps are coded in the RunOnceScript section using predefined variables – HOST?, IPADDRESS?, HOSTNAME, DOMAIN
- The Provisioning Component processes the XML specification
  - Provisions individual virtual machines.
  - Configures any dependencies between different nodes in the solution by executing commands in the RunOnceScript section

---

## Process of Activation

- Linked to Linux runlevels

Typical Linux runlevels

ID	Name	Description
1	Single-User Mode	Does not configure network interfaces, start daemons, or allow non-root logins. <sup>[3]</sup>
2	Multi-User Mode	Does not configure network interfaces or start daemons. <sup>[4]</sup>
3	Multi-User Mode with Networking	Starts the system normally. <sup>[5]</sup>
4	Not used/User-definable	For special purposes.
5	X11	As runlevel 3 + display manager.

---

## Process of Activation

- Add services to configure network and password to runlevel 3
  - Parameter file with name-value located on a floppy or cdrom attached to the VM
  - Service copies the parameter file to a folder
  - Invokes a service to take action

```
#!/etc/rc.status
#rc_reset

case "$1" in
  start)
    echo -e "Starting UAAE service: $SERUNAME"
    mount /dev/floppy /mnt
    cp /mnt/*vaap /opt/Activation/param
    umount /mnt
    # Remember status and be verbose
    sh $UAAE_BIN -s $SERUNAME
  # rc_status -v
  ;;
  stop)
    echo -e "Shutting down UAAE service: $SERUNAME"
  # rc_status -v
  ;;
  status)
    echo -e "Checking UAAE service: $SERUNAME"
  # rc_status -v
  ;;
  *)
    #if no parameters are given, print which are available.
    echo "Usage: $0 {start|stop|status|restart|reload}"
    exit 1

```

---

## Process of Activation

- Add services to configure network and password to runlevel 3
  - Parameter file with name-value located on a floppy or cdrom attached to the VM
  - Service copies the parameter file to a folder
  - Invokes a service to take action

```
#!/etc/rc.status
#rc_reset

case "$1" in
  start)
    echo -e "Starting UAAE service: $SERUNAME"
    mount /dev/floppy /mnt
    cp /mnt/*vaap /opt/Activation/param
    umount /mnt
    # Remember status and be verbose
    sh $UAAE_BIN -s $SERUNAME
  #
    rc_status -v
    ;;
  stop)
    echo -e "Shutting down UAAE service: $SERUNAME"
  #
    rc_status -v
    ;;
  status)
    echo -e "Checking UAAE service: $SERUNAME"
  #
    rc_status -v
    ;;
  *)
    #if no parameters are given, print which are available.
    echo "Usage: $0 {start|stop|status|restart|reload}"
    exit 1

```

---

## Parameter file

- Typical parameter file contains information shown below

```
<?xml version="1.0" encoding="UTF-8"?>
<vaap solution-name="ABC">
  <virtual-machine name="ABCsn">
    <software-resource name="NetworkConfig">
      <configuration aspect="Network">
        <parameter name="HostName" value="guest" />
        <parameter name="DomainName" value="cloud9.abc.com" />
        <parameter name="IPAddr" value="192.42.179.88" />
        <parameter name="NetMask" value="255.255.248.0" />
        <parameter name="Gateway" value="192.42.176.2" />
        <parameter name="DNSServer" value="192.4.250.46" />
      </configuration>
    </software-resource>
    <software-resource name="PasswordConfig">
      <configuration aspect="Security">
        <parameter name="root" value="gxUSx66il" />
      </configuration>
    </software-resource>
  </virtual-machine>
</vaap>
```



---

## Network configuration

- Network configuration script

```
BOOTPROTO=""static""
NETWORK=""
IPAddr=""$IPAddr""
NetMask=""$NetMask""

echo "DEVICE=eth0" > /tmp/ifcfg-eth0
echo "BOOTPROTO=$BOOTPROTO" >> /tmp/ifcfg-eth0
echo "IPADDR=$IPAddr" >> /tmp/ifcfg-eth0
echo "NETMASK=$NetMask" >> /tmp/ifcfg-eth0
echo "Type=Ethernet" >> /tmp/ifcfg-eth0
echo "ONBOOT=yes" >> /tmp/ifcfg-eth0
mv /tmp/ifcfg-eth0 $ROOT/etc/sysconfig/network-scripts/
echo "*      change IP address successfully"

#-----/etc/sysconfig/network routes -----
echo "NETWORKING=yes" > /tmp/network
echo "HOSTNAME=$FullName" >> /tmp/network
echo "GATEWAY=$Gateway" >> /tmp/network
mv /tmp/network /etc/sysconfig
route add default gw $Gateway

#echo "*      change gateway to $Gateway successfully"

#-----/etc/resolv.conf-----
echo "nameserver $DNS" > $ROOT/etc/resolv.conf
echo "search $Domain" >> $ROOT/etc/resolv.conf
echo "*      change DNS to $DNS successfully"
```

---

## Password configuration

- Password configuration script

```
while [ $# -ne 0 ]
do
  case $1 in
    root=*)
      Password=$1
      Password=`echo -e "$Password"|sed 's/root=//'`
      ;;
    *)
      ;;
  esac
  shift
done
echo "root:$Password" | chpasswd
```

---

## Post provisioning configuration

- After the initial VMs are configured, the Provisioning Manager needs to configure the software stack at each VM. This could be done through scripts that are run only once.

```
<runoncescript> (Executes this command on Node 0 VM)  
    "/root/fixup.sh $hostname $domain;  
    /opt/ibm/WebSphere/AppServer/setWAS.sh $HOST1;  
    /opt/ibm/WebSphere/AppServer/setDB2Server.sh $HOST2;"  
</runoncescript>
```

```
<runoncescript> (Executes this command on the Node 1 VM)  
    "/root/fixup.sh $hostname $domain;  
    /opt/ibm/WebSphere/AppServer/setDB2Server.sh $HOST2;"  
</runoncescript>
```

```
<runoncescript> (Executes this command on the Node 2 VM)  
    "/root/fixup.sh $hostname $domain;"  
</runoncescript>
```

---

## Summary

- Presented an overview of imaged based provisioning
- Activation process uses a scripting-based mechanism to configure virtual machines prior to boot up

---

## References

- Automating deployment and activation of virtual images
  - [http://www.ibm.com/developerworks/websphere/techjournal/0708\\_he/0708\\_he.html](http://www.ibm.com/developerworks/websphere/techjournal/0708_he/0708_he.html)
- Automating deployment and activation of virtual appliances for IBM AIX and Power Systems
  - [http://www.ibm.com/developerworks/websphere/library/techarticles/0904\\_ja/0904\\_ja.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0904_ja/0904_ja.html)