

Install IBM Cloud Private

This document is focused on the steps for installing IBM Cloud Private v2.1.0.3. The prerequisites section describes the assumptions for all that needs to have been done to prepare for the actual installation.

Prerequisites

1. A collection of virtual machines (VMs) has been provided for your use in this lab exercise. One of the machines is referred to as the "boot" machine, meaning it is the machine that is used to orchestrate the installation. The boot machine is also going to be the "master" node of the cluster. This node may also be referred to as the boot-master node. The other machines will be assigned the roles of "management", "vulnerability advisor (VA)", "proxy" and "worker".
2. DNS or the `/etc/hosts` file on each VM should be configured with the proper entries so that each VM can resolve the address of the other members of the cluster.
3. SSH needs to be set up such that the boot-master VM can ssh as root to each of the other VMs in the cluster (including itself) without using a password.
4. Ansible is installed on the boot-master machine. The `/etc/ansible/hosts` file has been configured to include the machines in your cluster. To check that Ansible is configured properly use: `ansible icp-cluster -m ping`. You should see a response from each machine including the boot-master.
5. The product install archive and the Docker install executable should be available on the boot-master machine in `/root/software/icp` and `/root/software/docker` respectively. (TBD - confirm this is where the boot VM will have the install bits) Normally, you would have downloaded the product archives from Passport Advantage (IBM customer) or eXtreme Leverage (IBM internal). You can find the GA releases by searching on, *IBM Cloud Private*.
6. The ICP Knowledge Center (KC) installation instructions for ICP Cloud Private "Cloud Native" or "Enterprise" are rooted in the section: [Intalling ICP Cloud Private Enterprise](#). (TBD - update this link to 2.1.0.3 KC once it is public)
7. On all VMs in the ICP cluster, if `firewalld` is running, stop it and disable it until after the ICP install completes. (Use the ansible playbook to stop and disable the firewall on each

VM.)

NOTE: The firewall only needs to be disabled during install. It gets enabled again on all members of the cluster after the install has completed.

NOTE In a scenario where an ICP cluster VMs (members) on more than one network segment/VLAN, then there may be physical firewalls that need to be configured to allow the ICP installation to proceed. See the ICP Knowledge Center section, [Default ports](#), (TBD - Update this link) for the list of ports that must be open for installation and configuration of an ICP instance.

Some additional boot-master pre-installation steps

This section has some steps that need to be taken on the boot-master before the actual installation command can be run.

NOTE: In these instructions, the root directory of the installation is referred to as `<ICP_HOME>`. A common convention is to install ICP in a directory that includes the ICP version in the directory name, e.g., `/opt/icp2103`.

- It is assumed that Docker is installed and running on the boot-master machine.
- It is assumed that the ICP images archive has been loaded into the Docker registry on the boot-master machine. (*NOTE:* The actual archive file name may be different depending on the version of ICP you are installing.)

```
tar -xf ibm-cloud-private-x86_64-2.1.0.3.tar.gz -O | docker load
```

- (On the boot-master) Extract the ICP boot meta-data to the `<ICP_HOME>/cluster` directory:

```
> cd <ICP_HOME>
> docker run -v $(pwd):/data -e LICENSE=accept ibmcom/icp-inception:2.1.0.3-ee cp -
```

NOTE: You may need to using a different version tag for the `icp-inception` image. Use `docker images | grep icp-inception` to see the version tag in your image repository.

The above command creates a directory named `cluster` in `<ICP_HOME>`. The `cluster` directory has the following contents:

```
> ls -l cluster
-rw-r--r--. 1 root root 3998 Oct 30 06:37 config.yaml
-rw-r--r--. 1 root root 88 Oct 30 06:37 hosts
drwxr-xr-x. 4 root root 39 Oct 30 06:37 misc
-r-----. 1 root root 1 Oct 30 06:37 ssh_key
```

- Add the IP address of all the cluster/cloud members to the `hosts` file in `<ICP_HOME>/cluster`.

NOTE: The ICP hosts file must use IP addresses. Host names are not used.

- Copy the ssh key file to the `/cluster`. (This overwrites the empty `ssh_key` file already there.)

```
> cp ~/.ssh/id_rsa ssh_key
cp: overwrite 'ssh_key'? y
```

- Check the permissions on the `ssh_key` file and make sure they are read-only for the owner (root). If necessary, change the permissions on the `ssh_key` file in `<ICP_HOME>/cluster` to "read-only" by owner, i.e., root.
- Check the access:

```
> ls -l ssh_key
-r-----. 1 root root 1675 Jun 30 13:46 ssh_key
```

- If the access is not read-only by owner, then change it:

```
> chmod 400 ssh_key
```

- Check again to make sure you changed it correctly.
- Copy/move the "image" archive (`ibm-cloud-private-x86_64-2.1.0.3.tar.gz`) to the images directory in `<ICP_HOME>/cluster`. (You first need to create the images directory.) In the command below it is assumed the image archive is located initially in `<ICP_HOME>`.

From `<ICP_HOME>/cluster`:

```
> mkdir images
> mv `<ICP_HOME>/ibm-cloud-private-x86_64-2.1.0.3.tar.gz` images
```

Working with the `config.yaml` file is described in the next section.

Configuring `config.yaml` on the boot-master

For information on the content of `config.yaml`, see the ICP KC section, [Cluster configuration settings](#).

For a simple sandbox deployment, the content of `config.yaml` can remain as is.

NOTE: The `network_cidr` and `service_cluster_ip_range` are set to "10." IP networks. If your cloud provide is using that same address range, then change the values to something else, e.g., some other "10." subnet or the "172.16." networks.

Things that can be left as-is for a small sandbox environment:

- `network_type` calico
- `network_cidr`: 10.1.0.0/16
- `service_cluster_ip_range`: 10.0.0.1/24
- For a simple cluster, everything else in `config.yaml` remains commented out.

There are many parameters that may be set in `config.yaml`. It is a good idea to read through the file to become familiar with the options.

Additional things that need to be set for a production environment:

- `vip_iface`, `cluster_vip`
- `proxy_vip_iface`, `proxy_vip`

You may want to include a `version` attribute in `config.yaml`. If you do, be sure it matches the version of the images in the docker registry that you want to use. You can do a `docker images` list to check the version tags of the available images. The version that will be deployed is set to an appropriate default in a YAML file in the `icp-inception` container so it setting the `version` value in `config.yaml` is intended for cases where the docker registry being used contains images from more than one version.

Likewise, you may want to include a `backup_version` attribute value in the `config.yaml`. Again, make sure the value of `backup_version` makes sense for the docker registry in use in that it matches the tag on the images that are intended to be the backup version.

NOTE: Gluster configuration in `config.yml` is not necessary when the Gluster servers are set up outside the ICP cluster, which is the topology used in this guide. **Do not** do any Gluster configuration in `config.yml`.

Other things that you may need to check

This section has a collection of items that have led to a failure in the installation process. This is a work in progress and is a place to keep track of this sort of stuff that seems a bit random.

- Make sure all the VMs in the cluster/cloud are running.

You may want to double check the following on each VM that is a member of the cluster:

- The network interface on each VM is started.
- The firewall on each VM is disabled.
- If you pre-installed Docker on each VM, then check that Docker is running on each VM.
- Docker must be installed and running on the boot-master VM.
- The ICP docker images must be loaded into the Docker registry on the boot-master VM.

Copy and load ICP Docker images tar ball to all cluster VMs

This section assumes that Docker is pre-installed on all of the cluster member VMs. Installing Docker on each VM uses the same steps as installing Docker on the boot-master VM.

NOTE: The need to do this step will likely go away with ICP releases starting with v2.1.0.2. The `icp-inception` process will configure a docker hub on the boot/master and the other nodes will pull images from it as needed. The local docker hub approach will make the image deployment process significantly more efficient.

It is expedient to pre-load the Docker registry on each VM that is a member of the ICP cluster. The installation process run from the boot-master machine will recognize that the Docker registry is up-to-date on the other cluster member machines and skip the step of copying and loading the image tar ball to the Docker registry on the given machine. The gain in the time it takes to load the registry is achieved because you can open as many shells as needed to do the copy and load operations concurrently.

- Copy the ICP image tar ball to all machines. (You can open multiple shells on the boot-master machine and start an `scp` of the image tar ball to each machine in the cluster.)
- Open a shell on each VM in the cluster and extract the docker images and load them into

the docker registry:

```
tar -xvf ibm-cloud-private-x86_64-2.1.0-beta-2.tar.gz -O | docker load
```

If you run out of file system space during the above command, use `df -h` to view file system utilization. You can use `df -ih` to view inode utilization. Make sure the file systems are adequately provisioned as described in the [Check the file system sizing](#Check the file system sizing) section above.

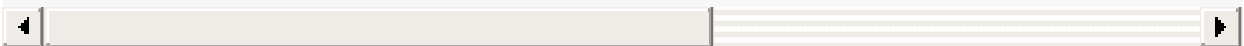
- On all but the boot-master machine, the ICP image tar file can be removed once the docker load completes. On the boot-master machine the ICP image tar file gets moved to an images directory in `<ICP_HOME>/cluster`.

NOTE: If you are building an ICP virtual machine, you can run the extract and load on the base VM image so that all cloned VMs have the docker registry pre-loaded with the ICP images.

No space left on device

During the load of the ICP images you may get something like:

```
Error processing tar file(exit status 1): write /opt/ibm/wlp/output/.classCache/C28
```



Do a `df -H` to see what is happening with file space.

Run the ICP install command

Docker is used to run the install for all members of the cluster/cloud. The command is shown below after some introductory notes. (This takes some time depending on the number of machines in the cluster. If you haven't pre-loaded the Docker images used by ICP, the image file gets copied to each VM and the images get loaded as part of the installation.) Run the install from `<ICP_HOME>/cluster` directory.

NOTE: It is assumed Docker is installed on the boot-master VM.

NOTE: It is assumed the ICP v2.1 images have been loaded into the local docker registry on the boot-master VM.

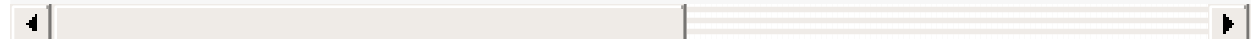
NOTE: In the docker commands below, `$(pwd)` is the current working directory of the shell where the command is run, i.e. `<ICP_HOME>/cluster`. It is assumed there are no space characters in the current working directory path. (It is a really bad idea to use space characters in directory and file names.) If you happen to have space characters in the current working directory path, then surround the `$(pwd)` with double quotes.

NOTE: It is OK to run this command multiple times to get things installed on all members of the cluster/cloud should problems show up with a particular cluster/cloud member. At least for basic problems, the error messages are very clear about where the problems are, e.g., network connectivity, firewall issues, docker not running.

NOTE: As of ICP v2.1.0.2, an uninstall is recommended after a failed installation. The installation detects that some lock files are still present if an uninstall has not been done and prompts you to do an uninstall. (Development work is on-going to make the ICP installation steps idempotent to avoid needing to do an uninstall after a failed installation.)

NOTE: During the installation all information messages go to stdout/stderr. If you want to capture a log of the installation process, you need to direct output to a file. The docker command line below uses `tee` to capture the log and also allow it to be visible in the shell window. A logs directory in `<ICP_HOME>/cluster` was created to hold the log files. The log file will have escape character sequences in it for color coding the text output, but it is readable.


```
> cd <ICP_HOME>/cluster
> mkdir logs
> docker run --net=host -t -e LICENSE=accept -v $(pwd):/installer/cluster ibmcom/icp
```

A terminal window with a light gray background and a dark gray border. It shows the execution of the docker run command. The command is: > docker run --net=host -t -e LICENSE=accept -v \$(pwd):/installer/cluster ibmcom/icp. The window has a scrollbar on the right side.

A common convention (not shown here) is to include a date stamp in the file name of the install log that gets written to /tmp (in this case) as well as a log number (in this case 1). The log number can be incremented each time the command is rerun if you want to save each log file.

NOTE: A single `-v` option is recommended to include a useful amount of trace information in the log. If you need to get more detail for installation problem determination purposes add a `-vv` or `-vvv` to the command line after the install verb for progressively more information, e.g.,

```
> docker run -e LICENSE=accept --net=host --rm -t -v $(pwd):/installer/cluster ibmcom/icp
```

A terminal window with a light gray background and a dark gray border. It shows the execution of the docker run command with -vvv. The command is: > docker run -e LICENSE=accept --net=host --rm -t -v \$(pwd):/installer/cluster ibmcom/icp. The window has a scrollbar on the right side.

- When the install completes, you want to see all "ok" and no "failed" in the recap. (The play

recap sample below is from a sandbox deployment. A production cluster will obviously have a lot more machines listed.)

```
PLAY RECAP *****
xxx.xx.xxx.50      : ok=109  changed=36  unreachable=0  failed=0
xxx.xx.xxx.52      : ok=109  changed=36  unreachable=0  failed=0
xxx.xx.xxx.57      : ok=137  changed=36  unreachable=0  failed=0
xxx.xx.xxx.60      : ok=163  changed=55  unreachable=0  failed=0
localhost          : ok=216  changed=114 unreachable=0  failed=0
```

- Problem determination is based on the installation log. The error messages are relatively clear. If the recap contains a non-zero failed count for any of the cluster members or something is unreachable, then grep/search the install log for `FAIL` or `fatal` to begin the problem determination process.
- Assuming the install went correctly move on to some basic "smoke tests" described in the section below.
- Once the installation completes successfully and you have performed some basic tests, the firewall on each cluster member will need to be configured to open ports as described in the ICP Knowledge Center section, [Default ports](#). Port 8443 on the master needs to be open to get to the ICP console.

Start and enable the firewalld on all cluster members

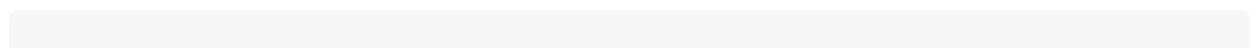
You may want to hold off on this step until some basic smoke tests have been executed. See the [\[Simple ICP smoke tests\]\(#Simple ICP "smoke" tests\)](#) section below.

After the install completes, configure firewall rules on each cluster member according to the ICP Knowledge Center section, [Default ports](#). Then start and enable firewalld on each cluster/cloud member.

Simple ICP "smoke" tests

This section documents some basic measures to confirm correct ICP operation.

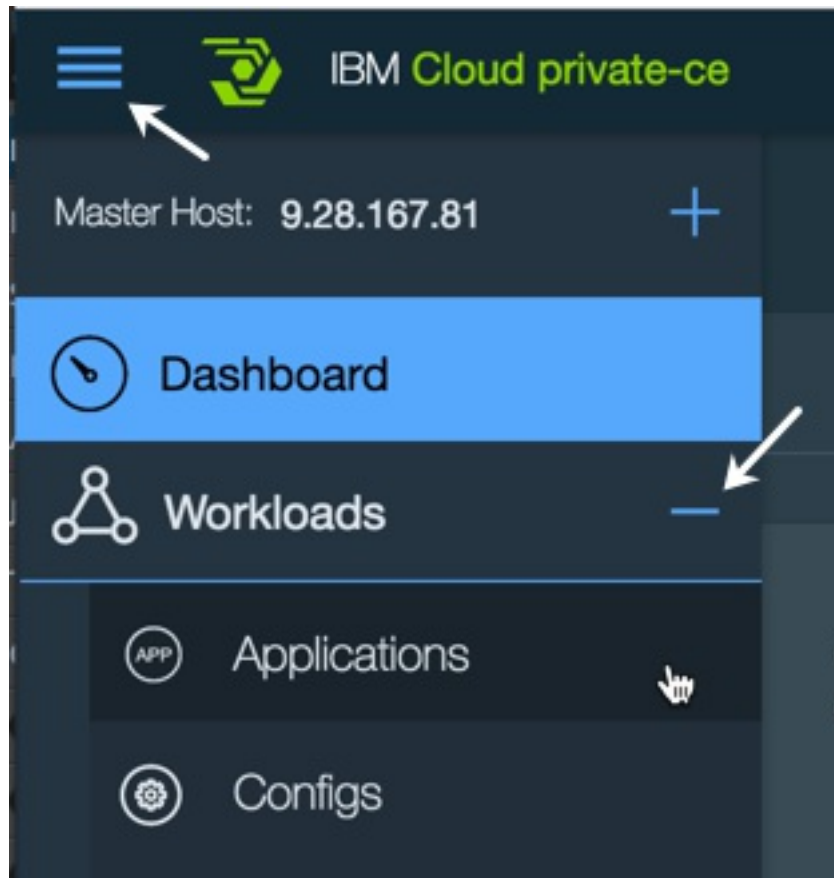
1. The simplest "smoke test" is to fire up the ICP admin console:




```
https://<boot_master>:8443/
```

Default user ID and password: admin/admin

2. Check that all processes are "available". In the ICP admin console you can see the workloads via the "hamburger" menu in the upper left corner margin. (See figure below.)



Troubleshooting installation issues

This section is a holding area for a collection of troubleshooting tips.

1. Install using the `-vvv` and piping output to tee to a log file is the first step. Examine the log file for the first sign of an error. Attempt the same command manually as the log indicates is having a problem to try to get to root cause.
2. Make sure the docker images repository has the correct images in it. For example, kubernetes-ee may not be present. Or the wrong version of kubernetes or some other image is in the docker images registry/repository.