PRJ-OCI-COMPUTE-100: Secure High-Performance Computing Cluster Implementation Guide

Author: Manus AI **Date:** January 26, 2026 **Project Folder:** `prj-oci-compute-100`

1. Project Overview

This document provides a comprehensive, production-ready implementation guide for deploying a **High-Performance Computing (HPC) Cluster** on Oracle Cloud Infrastructure (OCI). The core objective of this project is to establish a robust, scalable, and, most critically, a **secure-by-design** compute environment. Traditional HPC deployments often suffer from security vulnerabilities stemming from manual patching, the use of long-lived credentials, and exposed SSH ports. This solution directly addresses these challenges by integrating three fundamental OCI security and management services:

1. **OCI Instance Principals:** Implements a **zero-trust** authentication model, allowing compute instances to securely authorize against other OCI services (like Object Storage or OS Management) without requiring API keys, user credentials, or other long-lived secrets. This dramatically reduces the risk of credential theft and misuse.
2. **OCI Bastion Service:** Provides a secure, just-in-time, and ephemeral access mechanism to the private compute instances. By eliminating the need for public SSH exposure and acting as a managed jump host, the Bastion Service significantly shrinks the network attack surface.
3. **OCI OS Management Service (OSMS):** Ensures continuous compliance and security posture by automating the patching, configuration, and vulnerability management of the operating system. This moves the environment from a reactive to a proactive security model, ensuring that the HPC nodes are always up-to-date.

The resulting architecture is a highly secure, automatically managed fleet of compute resources, ideal for sensitive or regulated HPC workloads.

2. Business Context

The deployment of a secure and automated HPC cluster delivers substantial and quantifiable business value by mitigating operational risks and improving efficiency. The primary driver for this architecture is the need to scale compute resources rapidly while maintaining a stringent security and compliance posture, which is often a bottleneck in traditional environments.

Section	Description and Quantified Value
The Problem	Large-scale compute environments are highly vulnerable to attacks, misconfigurations, and unauthorized access. Manual security processes, such as patching and configuration management, do not scale, leading to significant operational overhead and increased risk exposure. A single unpatched vulnerability can lead to a costly security incident.
The Solution	A secure OCI compute implementation leveraging automated patching (OSMS), secure access (Bastion), and zero-trust authentication (Instance Principals). This approach embeds security into the infrastructure lifecycle.
Business Value (ROI & Efficiency)	**Risk Reduction (ROI):** Eliminating public SSH exposure and long-lived credentials can reduce the likelihood of a major breach by an estimated **70-80%**. **Operational Efficiency:** Automated patching and configuration via OSMS can save up to **10-15 hours per month per 100 instances** in manual administrative effort. **Compliance Cost Savings:** Automated audit evidence generation (Bastion logs, patch reports) significantly reduces the time and cost associated with regulatory audits.
Risk Mitigation	This architecture prevents unauthorized network access, exploitation of unpatched software vulnerabilities, credential theft, and configuration drift across the compute fleet. It ensures that the compute resources remain in a known, secure state, which is critical for maintaining data integrity and availability.

The move to a zero-trust, automated management model translates directly into a lower Total Cost of Ownership (TCO) for the HPC environment, primarily through reduced security incident response costs and optimized IT staff utilization.

3. GRC Mapping (Governance, Risk, and Compliance)

This secure HPC deployment is engineered to align with several major Governance, Risk, and Compliance (GRC) frameworks. The implementation of OCI services directly maps to specific security controls required by these standards, providing clear audit evidence.

Control Category	OCI Service Implementation	GRC Framework Mapping
Vulnerability Management	**OS Management Service (OSMS):** Automates the identification and deployment of security patches and updates.	**NIST CSF:** PR.IP-12 (Vulnerability management). **ISO 27001:** A.12.6 (Technical vulnerability management). **PCI DSS:** Requirement 6.2 (Patching).
Secure Access & Remote Management	**OCI Bastion Service:** Provides ephemeral, just-in-time, and auditable access to private resources, eliminating persistent public exposure.	**NIST 800-53:** AC-17 (Remote Access). **SOC 2:** CC6.1 (Logical access controls). **HIPAA:** § 164.308(a)(5)(ii)(B) (Protection from malicious software/unauthorized access).
Identity and Authentication	**Instance Principals (Dynamic Groups & IAM Policy):** Implements a zero-trust model by granting services access based on instance identity, not static credentials.	**NIST 800-53:** IA-2 (Identification and Authentication). **GDPR:** Article 32 (Security measures, including access control).
Configuration Management	**Cloud-Init & OSMS:** Enforces initial security hardening (e.g., disabling root SSH) and maintains a secure baseline configuration.	**CIS Benchmarks:** OS hardening standards. **NIST CSF:** PR.IP-1 (Baseline configuration). **PCI DSS:** Requirement 2 (Configuration standards).

Audit Evidence: The OCI platform automatically generates the necessary audit trails:

- **Patch Compliance Reports:** Provided by OS Management Service, detailing the patch status of all instances.
- **Bastion Session Logs:** Detailed records of all secure access sessions, including user, time, and duration.

- **IAM Policy Configuration:** Demonstrates the principle of least privilege for instance-to-service communication.

4. Prerequisites

Successful deployment requires the following OCI resources and configurations to be established prior to executing the implementation steps. All placeholder OCIDs (`ocid1.compartment.oc1..aaaaaaexample`, etc.) must be replaced with your actual resource identifiers.

Prerequisite	Description	Best Practice & Requirement
OCI Account & Permissions	An active OCI tenancy with necessary IAM policies to create/manage Compute, Networking, IAM, and Bastion resources.	Ensure the deploying user belongs to a group with `manage all-resources` permissions in the target compartment.
Compartment	A dedicated compartment to logically isolate the HPC cluster resources.	Use a clear naming convention, e.g., `cmp-hpc-compute-prod`.
Virtual Cloud Network (VCN)	A VCN with sufficient CIDR block space to host the cluster.	VCN should be configured with a Service Gateway to allow private instances to reach OCI public services (like OSMS) without traversing the internet.
Private Subnet	The subnet where the HPC compute instances will be launched. **Crucially, this subnet must not have a Route Table entry to an Internet Gateway.**	Security Lists must be configured to allow inbound SSH (port 22) only from the Bastion Service's subnet and outbound access to the Service Gateway.
Public Subnet (for Bastion)	A separate subnet to host the OCI Bastion service endpoint. This subnet requires a Route Table entry to an Internet Gateway.	Use a small CIDR block, as only the Bastion endpoint will reside here.
SSH Key Pair	An SSH key pair (public and private) for accessing the instances via the Bastion service.	The public key is used during instance launch and Bastion session creation. The private key is required for the final SSH connection.
OCI CLI	The Oracle Cloud Infrastructure Command Line Interface must be installed and configured on the local machine for executing the deployment commands.	Ensure the CLI is authenticated with an API key and configured to the correct region.

5. Architecture Overview

The architecture is centered around a secure, multi-layered defense model for the HPC compute fleet.

1. **Compute Instances:** OCI Compute instances are deployed into a **Private Subnet**. They are assigned **no public IP address**, making them unreachable from the public internet.
2. **Zero-Trust Identity:** Each instance is a member of an **IAM Dynamic Group** based on its compartment. This group is granted an **IAM Policy** that allows the instances to act as **Instance Principals**. This identity is used to communicate with OCI services, specifically OS Management and the Bastion Service, for registration and session creation.
3. **Secure Access Layer:** The **OCI Bastion Service** is deployed into a **Public Subnet**. It acts as a managed proxy. When a user needs access, they create a time-limited **Bastion Session** targeting the private IP of the compute instance. The Bastion then facilitates a secure SSH tunnel, eliminating the need for direct SSH access.
4. **Continuous Management:** The **OS Management Service (OSMS)** agent, enabled via `cloud-init`, registers the instance with OSMS. This ensures that the instance is continuously monitored for patch compliance and security configuration drift, fulfilling the GRC requirements for vulnerability management.

This design ensures that the instances are secure by default, with access being ephemeral, auditable, and based on the principle of least privilege.

6. Step-by-Step Implementation

The deployment is executed using the OCI Command Line Interface (CLI). It is recommended to define all variables in a shell script before execution.

Setup: Define Environment Variables

Before running any commands, define the following variables with your specific OCIDs and values.

```
# Define your OCI environment variables
COMPARTMENT_OCID="ocid1.compartment.oc1..aaaaaaexample_hpc_compartment"
REGION="us-ashburn-1"
BASTION_SUBNET_OCID="ocid1.subnet.oc1..aaaaaaexample_public_subnet_bastion"
PRIVATE_SUBNET_OCID="ocid1.subnet.oc1..aaaaaaexample_private_subnet_hpc"
IMAGE_OCID="ocid1.image.oc1..aaaaaaexample_os_image" # Use a secure,
hardened image (e.g., Oracle Linux 8/9)
SHAPE="VM.Standard3.Flex"
SSH_KEY_FILE="~/.ssh/id_rsa.pub" # Path to your public SSH key
```

Step 1: Configure IAM for Zero-Trust Access

This step establishes the identity for the compute instances, allowing them to securely interact with OCI services.

1.1. Create a Dynamic Group (DG)

The DG automatically includes all compute instances launched in the target compartment, enabling a scalable identity solution.

```
oci iam dynamic-group create \
    --name "DG-PRJ-OCI-COMPUTE-100" \
    --description "Dynamic Group for PRJ-OCI-COMPUTE-100 instances to use
Instance Principals" \
    --matching-rule "All {instance.compartment.id = '$COMPARTMENT_OCID'}" \
    --region "$REGION"
```

1.2. Create an IAM Policy

This policy grants the Dynamic Group the minimum required permissions to use OS Management and create Bastion sessions.

```
oci iam policy create \
    --name "Policy-PRJ-OCI-COMPUTE-100" \
    --description "Policy for PRJ-OCI-COMPUTE-100 Dynamic Group to use OS
Management and Bastion" \
    --statements '["Allow dynamic-group DG-PRJ-OCI-COMPUTE-100 to use osms-
managed-instances in tenancy", "Allow dynamic-group DG-PRJ-OCI-COMPUTE-100
to manage bastion-sessions in tenancy"]' \
    --compartment-id "$COMPARTMENT_OCID" \
    --region "$REGION"
```

Step 2: Deploy the OCI Bastion Service

The Bastion service is the single, secure entry point for administrative access.

2.1. Create the Bastion Resource

The Bastion endpoint is deployed into the designated public subnet. The `max-session-ttl` is set to 3 hours (10800 seconds) for security.

```
oci bastion bastion create \
    --bastion-type "STANDARD" \
    --compartment-id "$COMPARTMENT_OCID" \
    --target-subnet-id "$BASTION_SUBNET_OCID" \
    --name "Bastion-PRJ-OCI-COMPUTE-100" \
    --max-session-ttl 10800 \
    --phone-book-entry "ssh-user@hpc-cluster" \
    --region "$REGION"
```

Note: Wait for the Bastion resource to become active before proceeding. You can query its state using `oci bastion bastion get --bastion-id <BASTION_OCID>`.

Step 3: Launch the Secure Compute Instance

The instance is launched in the private subnet with a `cloud-init` script for initial hardening and agent setup.

3.1. Prepare the Cloud-Init Script

Create a file named `cloud-init.yaml` with the following content. This script updates packages, sets up a non-root user (`opc`), disables root SSH login, and ensures the necessary OCI agents are running. **Replace `<YOUR_PUBLIC_SSH_KEY>` with the actual content of your public key.**

```
#cloud-config
package_update: true
package_upgrade: true
users:
  - name: opc
    ssh_authorized_keys:
      - <YOUR_PUBLIC_SSH_KEY>
    sudo: ALL=(ALL) NOPASSWD:ALL
    groups: users, wheel
runcmd:
  - [ sh, -c, "sed -i 's/^PermitRootLogin yes/PermitRootLogin no/'
/etc/ssh/sshd_config" ]
  - [ sh, -c, "systemctl restart sshd" ]
  - [ sh, -c, "systemctl enable --now oracle-cloud-agent" ]
  - [ sh, -c, "systemctl enable --now osmanagementhub-agent" ]
```

3.2. Launch the Instance

The instance is launched without a public IP (`--assign-public-ip false`) and uses the base64-encoded `cloud-init.yaml` as `user_data`.

```
# Base64 encode the cloud-init file for the metadata
USER_DATA_BASE64=$(base64 -w 0 cloud-init.yaml)

oci compute instance launch \
    --availability-domain "AD-1" \
    --compartment-id "$COMPARTMENT_OCID" \
    --display-name "HPC-Node-01" \
    --image-id "$IMAGE_OCID" \
    --shape "$SHAPE" \
    --subnet-id "$PRIVATE_SUBNET_OCID" \
    --assign-public-ip false \
    --metadata "{\"ssh_authorized_keys\": \"$(cat $SSH_KEY_FILE)\",
\"user_data\": \"$USER_DATA_BASE64\"}" \
    --region "$REGION"
```

Step 4: Access the Instance via Bastion Session

Access is granted by creating a temporary, managed SSH session.

4.1. Retrieve Resource OCIDs and IP

Wait for the instance to be provisioned, then retrieve the necessary identifiers.

```
INSTANCE_OCID=$(oci compute instance list --compartment-id
"$COMPARTMENT_OCID" --display-name "HPC-Node-01" --query "data[0].id" --raw-
output)
INSTANCE_IP=$(oci compute instance list --compartment-id "$COMPARTMENT_OCID"
--display-name "HPC-Node-01" --query "data[0].'private-ip'" --raw-output)
BASTION_OCID=$(oci bastion bastion list --compartment-id "$COMPARTMENT_OCID"
--display-name "Bastion-PRJ-OCI-COMPUTE-100" --query "data[0].id" --raw-
output)

echo "Instance OCID: $INSTANCE_OCID"
echo "Instance Private IP: $INSTANCE_IP"
echo "Bastion OCID: $BASTION_OCID"
```

4.2. Create a Managed SSH Session

This command creates a session that tunnels SSH traffic through the Bastion to the private IP of the instance. The session is valid for 30 minutes (1800 seconds).

```
# Create a Bastion Session
SESSION_OCID=$(oci bastion session create-managed-ssh-session \
    --bastion-id "$BASTION_OCID" \
    --target-resource-id "$INSTANCE_OCID" \
    --target-resource-private-ip "$INSTANCE_IP" \
    --key-type "PUB" \
    --key-details "$(cat $SSH_KEY_FILE)" \
    --session-ttl 1800 \
    --display-name "SSH-Session-HPC-Node-01" \
    --query "data.id" --raw-output)

echo "Bastion Session OCID: $SESSION_OCID"
```

4.3. Connect to the Instance

Wait for the session to become active, then retrieve the dynamically generated SSH command and execute it.

```
# Wait for the session to be active (may take a minute)
# ...

# Get the SSH connection string
SSH_COMMAND=$(oci bastion session get --session-id "$SESSION_OCID" --query
"data.\"ssh-details\".port-forwarding-session-ssh-details.ssh-command" --
raw-output)

# Execute the command to connect
# Note: This command must be run in your local terminal, not within the OCI
CLI environment.
echo "To connect, run the following command locally:"
echo "$SSH_COMMAND"
```

7. Validation & Testing

After deployment, a series of validation steps must be performed to ensure all security and management components are functioning correctly.

7.1. Verify Bastion Access

The most fundamental test is to successfully connect to the instance using the generated SSH command from Step 4.3.

- **Expected Result:** A successful SSH connection to the instance using the `opc` user, confirming that the instance is running, the SSH daemon is active, and the Bastion tunnel is functional.
- **Verification:** Once connected, verify the `cloud-init` hardening by attempting to switch to the root user via SSH (which should be denied) and checking the `sshd_config` file.

7.2. Verify Instance Principal Authentication

This confirms that the Dynamic Group and IAM Policy are correctly configured, allowing the instance to authenticate with OCI services.

- **Execution:** From within the connected instance, execute a simple OCI CLI command using the `instance_principal` authentication method.

```
# Test Instance Principal
oci iam compartment list --all --auth instance_principal
```

- **Expected Result:** The command should successfully return a list of compartments visible to the instance's IAM policy. A failure indicates an issue with the Dynamic Group matching rule or the IAM Policy statements.

7.3. Verify OS Management Registration

This confirms that the OSMS agent is running and the instance is registered for continuous patching and compliance.

- **Execution:** Check the OCI Console under **Compute -> Instances** or use the CLI to query the managed instance status.

```
# Check if the instance is a managed instance
oci os-management-hub managed-instance get --managed-instance-id
"$INSTANCE_OCID"
```

- **Expected Result:** The output should show the instance details, including its lifecycle state and its status as a managed instance. The OCI Console should show the instance as “Managed” and report its patch status.

8. Troubleshooting

Issue	Potential Cause	Resolution
Bastion Session Fails to Connect	**Networking:** Security List on the private subnet is blocking inbound SSH (port 22) from the Bastion subnet. **Session TTL:** The session has expired (max 3 hours).	**Networking:** Update the private subnet's Security List to allow ingress on port 22 from the Bastion subnet's CIDR. **Session TTL:** Create a new Bastion session.
Instance Principal Test Fails	**Dynamic Group Rule:** The matching rule in the Dynamic Group is incorrect (e.g., wrong compartment OCID). **IAM Policy:** The policy statement is misspelled or missing the required permissions (`use osms-managed-instances`).	**Dynamic Group:** Verify the `instance.compartment.id` matches the instance's compartment. **IAM Policy:** Ensure the policy statements are syntactically correct and grant the necessary verbs/resource types.
OS Management Agent Not Reporting	**Service Gateway:** The private subnet lacks a Service Gateway or the Route Table is missing a route to the Service Gateway. **Agent Failure:** The `osmanagementhub-agent` service failed to start.	**Networking:** Verify the private subnet's Route Table has a route for the OCI services CIDR block via the Service Gateway. **Agent:** SSH into the instance (via Bastion) and check the agent status: `systemctl status osmanagementhub-agent` .
Instance Launch Fails	**Cloud-Init Error:** Syntax error in `cloud-init.yaml` or the base64 encoding failed. **Shape/Image:** The specified shape or image is unavailable in the selected Availability Domain.	**Cloud-Init:** Check the instance's console history for `cloud-init` logs. Ensure the `user_data` is correctly base64 encoded. **Shape/Image:** Verify availability in the OCI Console.

9. Cost Optimization

Optimizing the cost of an HPC cluster is critical due to the high resource consumption. This architecture allows for several key cost-saving measures:

- **Flexible Shapes (`VM.Standard3.Flex`):** By utilizing flexible shapes, you can customize the number of OCPUs and the amount of memory independently. This ensures you only provision and pay for the exact resources required for the HPC workload, avoiding the waste associated with fixed-size shapes.
- **Autoscaling and Instance Pools:** For burstable or time-sensitive HPC workloads, implement an **Autoscaling Configuration** on an Instance Pool. This allows the cluster to automatically scale down the number of compute nodes to zero or a minimum baseline during off-peak hours, significantly reducing compute costs.
- **Boot Volume Size:** Use the minimum required boot volume size (typically 50GB for OS) and rely on separate, cost-effective block or file storage for the actual HPC data and scratch space. Avoid over-provisioning the boot volume.
- **Preemptible Instances:** For fault-tolerant or non-critical HPC jobs, consider using **Preemptible Instances** (if available for the chosen shape) which offer substantial discounts in exchange for the possibility of termination.
- **Networking Costs:** While the Bastion service itself is free, the associated public subnet resources (NAT Gateway, Public IP) incur standard networking costs. Ensure the NAT Gateway is only used for necessary outbound traffic (like patching and OSMS communication) and is not a general-purpose internet egress point.

10. Security Best Practices

The security-focused architecture already implements several best practices, but a production environment requires continuous adherence to these principles.

- **Principle of Least Privilege (PoLP):** The IAM policy for the Dynamic Group is strictly scoped to the necessary services (`osms-managed-instances` , `bastion-sessions`). This PoLP must be maintained for any future extensions of the instance's permissions.
- **Ephemeral Access:** Enforce the use of the OCI Bastion Service for all administrative access. **Never** enable public IP addresses or direct SSH access. Keep Bastion session Time-To-Live (TTL) as short as possible (e.g., 30 minutes to 1 hour) and require users to create a new session for each access event.
- **Automated Hardening:** The `cloud-init` script provides an initial security baseline (disabling root SSH, setting up a non-root user). This script should be

regularly reviewed and updated to reflect the latest CIS Benchmarks for the operating system.

- **Continuous Patching and Monitoring:** Rely entirely on the OS Management Service for all patching. Configure OSMS to automatically apply critical and security patches. Additionally, integrate OCI Monitoring and Logging Analytics to track agent health, login attempts, and system events.
- **Network Segmentation:** Ensure the HPC private subnet is completely isolated from other production or corporate networks, only allowing necessary traffic (e.g., NFS/storage traffic) via Network Security Groups (NSGs) or Security Lists.
- **Data Encryption:** All data at rest (Boot Volumes, Block Volumes, File Storage) should be encrypted using OCI Vault keys, ensuring compliance with data protection regulations.

11. Cleanup

To fully decommission the project and avoid future charges, execute the following steps in reverse order of deployment.

11.1. Terminate the Compute Instance

```
oci compute instance terminate --instance-id "$INSTANCE_OCID" --force
```

11.2. Delete the Bastion Service

```
oci bastion bastion delete --bastion-id "$BASTION_OCID" --force
```

11.3. Delete the IAM Policy

You will need the OCID of the policy created in Step 1.2.

```
# Replace with the actual policy OCID
oci iam policy delete --policy-id "ocid1.policy.oc1..aaaaaaexample_policy" --force
```

11.4. Delete the Dynamic Group

You will need the OCID of the dynamic group created in Step 1.1.

```
# Replace with the actual dynamic group OCID
oci iam dynamic-group delete --dynamic-group-id "ocid1.dynamicgroup.oc1..aaaaaaexample_dg" --force
```

Document Word Count (Approximate): 3,500 words.