TEMPLATE

VMware Virtual Machine RBAC Design Template

A practical template for defining least-privilege role-based access control for VMware virtual machines, including role scope, permissions, approvals, validation, and review.

VMware Virtual Machine RBAC Design Template

Use this template to design, assign, validate, and review role-based access control for VMware virtual machines. It is intentionally vendor-neutral in structure while remaining operationally useful for VMware environments.

---

1) Purpose

Environment: _____

Application / workload: _____

Owner: _____

Date created: _____

Review cycle: _____

Document version: _____

Objective

Define least-privilege access so users can only perform the VM actions required for their role.

Security goals

- Roles are based on tasks, not job titles.
- Permissions are granted at the narrowest practical scope.
- Privileged actions are separated where possible.
- Access can be validated, audited, and reviewed.

---

2) Preconditions

Before applying permissions, confirm:

- [ ] Inventory boundaries are clean and current.
- [ ] Ownership is known for each VM, folder, or resource pool.
- [ ] The admin boundary has been defined.
- [ ] Console access approvals are documented, if required.
- [ ] Guest OS access approvals are documented, if required.
- [ ] A non-production test target is available.
- [ ] Identity groups or users to be integrated are known.
- [ ] A rollback plan exists for removing roles.
- [ ] Built-in and custom role usage has been decided.
- [ ] You have permission to view assignments and audit results.

---

3) Role design matrix

Complete one row for each role.

Role name	Business task	Allowed actions	Scope	Approval owner	Notes
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

Role design rules

- [ ] Permissions support an actual operational workflow.
- [ ] Each role has a clear owner.
- [ ] Scope is no broader than necessary.
- [ ] Admin-like access is not bundled into convenience roles.
- [ ] Read-only access is separated from control access where possible.

---

4) Scope map

Document where each role applies.

Inventory object	Direct assignment	Inherited assignment	Excluded from scope	Reason
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Scope guidance

- [ ] Use individual VM scope only for highly sensitive workloads.
- [ ] Use folders for application- or environment-level boundaries.
- [ ] Use resource pools only when compute boundaries are the right control.
- [ ] Use datacenter-level scope only for intentionally broad read-only access.
- [ ] Review inherited access before relying on it.

---

5) Role definition worksheet

Complete for each custom or built-in role you plan to use.

Role name

_____

Purpose

_____

Included permissions

- _____
- _____
- _____
- _____

Explicitly excluded permissions

- _____
- _____
- _____
- _____

Scope

_____

Assignment method

- [ ] Directory group
- [ ] Individual account
- [ ] Break-glass account
- [ ] Other: _____

Justification

_____

_____

Risk review

If this account were compromised, the maximum likely impact would be:

_____

Mitigations:

- _____

- _____
- _____

---

6) Recommended role categories

Use these as a starting point and tailor them to your environment.

VM operator

Task: Power on/off, open console, basic operational checks.

Should not include: Configuration changes, snapshot creation, cloning.

Application owner

Task: View VM status, notes, and operational health.

Should not include: Console access unless formally approved.

Backup operator

Task: Snapshot- or backup-related actions on approved workloads.

Should not include: Console access or edit settings unless required.

Security reviewer

Task: Read-only visibility into inventory, tasks, and events.

Should not include: Operational changes.

Support engineer

Task: Limited troubleshooting access for assigned VMs.

Should not include: Broad administrative access.

---

7) Group assignment plan

Assign roles to groups whenever possible.

Group name	Role assigned	Member type	Owner	Review cadence	Notes
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

Group controls

- [ ] Group purpose is obvious from the name.
- [ ] Group membership is reviewed regularly.
- [ ] Temporary access is time-boxed.
- [ ] Direct user assignments are minimized.
- [ ] Emergency access is separate from standard access.

---

8) Validation checklist

Validate access on a non-production target before rollout.

Effective access test

For each role, confirm the user can:

- [ ] View the intended VM or scope.
- [ ] Perform allowed power actions.
- [ ] Open console only when approved.
- [ ] Edit settings only when approved.
- [ ] Create or manage snapshots only when approved.
- [ ] Clone only when approved.
- [ ] See tasks and events if required.
- [ ] Access nothing outside the intended scope.

Validation log

Test case	Expected result	Actual result	Pass / fail	Notes
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Common failure checks

- [ ] Parent-folder access does not expose unrelated VMs.
- [ ] Inherited permissions are understood.
- [ ] Broad roles were not created for convenience.

- [ ] Read-only access stays read-only.
- [ ] Console access is not bundled without approval.

---

9) Audit and review plan

Review cadence

Frequency: _____

Review owners

- Identity owner: _____
- Application owner: _____
- Infrastructure owner: _____
- Security reviewer: _____

Review questions

- [ ] Does each role still support a real task?
- [ ] Are permissions still the narrowest practical scope?
- [ ] Have any groups grown beyond their purpose?
- [ ] Are there direct assignments that should be converted to groups?
- [ ] Do any users retain access after job changes or project completion?
- [ ] Have any exceptions become permanent?

Evidence to retain

- [ ] Role matrix
- [ ] Scope map
- [ ] Group membership exports
- [ ] Validation results
- [ ] Exception approvals
- [ ] Review sign-off

---

10) Rollback plan

If access must be removed quickly, document the steps below.

Rollback trigger

_____

Immediate actions

- _____
- _____
- _____

Recovery notes

_____

_____

Escalation contact

_____

---

11) Implementation sign-off

Prepared by

Name: _____

Signature: _____

Date: _____

Reviewed by

Name: _____

Signature: _____

Date: _____

Approved by

Name: _____

Signature: _____

Date: _____

---

12) Quick reference for ongoing operations

Keep

- [ ] Task-based roles
- [ ] Narrow scopes
- [ ] Group-based assignments
- [ ] Regular access reviews
- [ ] Validation after changes

Avoid

- [ ] Title-based privilege design
- [ ] Broad parent-folder access without review
- [ ] Permanent exceptions without justification
- [ ] Direct user assignments for routine access
- [ ] Roles that quietly become delegated admin

Final check

Before production rollout, confirm:

- [ ] The role matrix is complete.

- [ ] Scope is documented and intentional.
- [ ] Validation has passed on a test target.
- [ ] Review ownership is assigned.
- [ ] Rollback steps are ready.
- [ ] Evidence will be retained for audit.

Result: This RBAC design is ready for controlled production implementation.